12

CR 93.003

NCEL

June 1993

Contract Report

An Investigation Conducted by
Adaptive Research Corporation
Huntsville, AL

STRUCTURED FINITE VOLUME MODELING OF U.S. NAVY AIRCRAFT ENGINE TEST CELLS

TASK 2: TURBOPROP ENGINE -

CODE DOCUMENTATION AND LISTINGS - VOLUME 2

Abstract This report presents results of the numerical simulation of a U.S. Naval turboprop test cell facility. The ultimate purpose of this simulation was to provide the Navy with a numercial model to be used for the evaluation of the aerothermal performance of test cells. This simulation was performed using the structured finite volume (SFV) computer code. A description of the physical model, mathematical details, boundary conditions, and results of the study are presented and covered in this report.

Volume 2, Code Documentation and Listings, provides a copy of the input files developed for the modeling of turboprop test cells.

14885

NAVAL CIVIL ENGINEERING LABORATORY PORT HUENEME CALIFORNIA 93043-4328

Approved for public release; distribution is unlimited.

METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

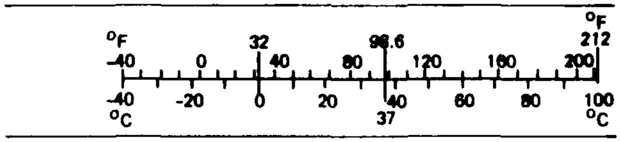
Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
in	inches	*2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
		AREA		
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
		MASS (weight)		
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2,000 lb)	0.9	tonnes	t
		VOLUME		
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
		TEMPERATURE (exact)		
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

*1 in = 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures, Price $2.25, SD Catalog No. C13.10:286.

inches
1
2
3
4
5
6
7
8
9
cm
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
		AREA		
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
		MASS (weight)		
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1,000 kg)	1.1	short tons	
		VOLUME		
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
		TEMPERATURE (exact)		
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-018

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE

June 1993

3. REPORT TYPE AND DATES COVERED

Final; December 1990 - September 1992

4. TITLE AND SUBTITLE STRUCTURED FINITE VOLUME MODELING OF U.S. NAVY AIRCRAFT ENGINE TEST CELLS TASK 2: TURBOPROP ENGINE - CODE DOCUMENTATION AND LISTINGS - VOLUME 2

5. FUNDING NUMBERS

PR - 0604215N
C - N47408-91-C-1228
DN - 661008

6. AUTHOR(S)

P. L. Daley and W. A. Mahaffey

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESSE(S)

Adaptive Research Corporation
4960 Corporate Drive, Suite 100-A
Huntsville, AL 35805

8. PERFORMING ORGANIZATION REPORT NUMBER

CR 93.003

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESSE(S)

Commander / Naval Civil Engineering Laboratory
Naval Air Systems Command 560 Laboratory Drive
Code 09Y Facilities Systems Division/L53
Washington, DC 20362-5101 Port Hueneme, CA 93043-4328

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report presents results of the numerical simulation of a U.S. Naval turboprop test cell facility. The ultimate purpose of this simulation was to provide the Navy with a numercial model to be used for the evaluation of the aerothermal performance of test cells. This simulation was performed using the structured finite volume (SFV) computer code. A description of the physical model, mathematical details, boundary conditions, and results of the study are presented and covered in Volume 1.

Volume 2, Code Documentation and Listings, provides a copy of the input files developed for the modeling of turboprop test cells.

14. SUBJECT TERMS

Computational fluid dynamics, test cells, aviation test facilities

15. NUMBER OF PAGES

59

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UL

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1 Purpose of the Report

This report provides a copy of the input files developed for the modeling of turboprop test cells. These copies are contained in the Appendices of this report and are described briefly below. A detailed discussion of building a computational grid for this project is provided in the second section of this report. The results of the turboprop test cell modeling are reported in the first volume of this report.

1.2 The Listings Provided

The listings are contained in Appendices B through D. Appendix B contains the Q1 input file, Appendix C contains the FORTRAN SATELLITE program, and Appendix D contains the FORTRAN GROUND file. Sketches are provided in Appendix A.

2. USER SECTION

2.1 Grid Generation

In this section a detailed discussion for the creation of a computational grid is supplied. The bulk of the input for this model deals with producing a computational grid. The code was designed for relatively easy modifications with the flexibility to model a range of changes as called for in the scope of work.

The premise of this procedure is that a 2–dimensional package will be used to create various cross sectional planes. These planes will then be stacked, blended or rotated to create the final 3–dimensional computational grid. In general, the program works as follows: 1.) the user specifies all the inputs necessary for the creation of all the various 2–dimensional cross sectional (X–Y) planes inside the standard input files (Q1 and SATLIT), 2.) the standard input files are then executed to produce the data files needed for the 2–dimensional grid generation program (EasyMesh2D or GGP), 3.) GGP is then executed for each data plane produced, and 4.) the standard input files are re–executed to produce the final grid and the other input files needed for the solver.

The standard input files will create 5 types of X–Y planes. Each plane can have several different varieties or subsets. The first type (TYPE 1) of plane is used to describe the test bed up to the engine. The planes are broken down into various regions in the X and Y directions. The user must specify the total distance from the origin for each region, the number of cells in each region, and the clustering factor for the griding of each region. Each of these will be detailed later in this section.

The second type (TYPE 2) is used to describe the X–Y cross section of the engine exit and the augmenter lip. TYPE 3 is used for X–Y cross section that across the augmenter tube. The fourth type (TYPE 4) is used to describe the triangular room in front of the chimney and the front face of the chimney. The final type (TYPE 5) is used to describe the exit plane. Additional information may be supplied in the input files.

The file name nomenclature for the data files for the GGP is that the file name starts with the letter CS. Then numbers are added as suffixes starting at 61 and continuing until all planes are created. The data files are created in order. For TYPE 1 there are five different subsets (CS files) created. The first (CS61) is used to describe the inlet plane, the second (CS62) produces a cross section of the front of the orifice while the third (CS63) produces the back of the orifice, the fourth (CS64) represents a X–Y section across the reduction gear, and the last (CS65) is used to describe the engine inlet.

CS61 is a mostly orthogonal grid used to represent the inlet plane. Various lines will be converted to arcs in order to represent the orifice, prop, reduction gear, and engine inlet. CS62 has an outer circle which represents the orifice. It also contains two other circles, which do not physically represent an object at this plane but will be used in other cross sections to represent the prop (middle circle) or the reduction gear or engine inlet (inside circle). This procedure helps to maximize the orthogonality for the total grid. CS63 is identical to CS62 with the exception that the diameter of the orifice has been reduced. CS64 is a repeat of CS62 thus allowing the spacing between the outer and middle circle to be expanded. The final cross section (CS65) is identical to CS64 except that in inner circle it now represents the engine inlet.

For TYPE 1 files there are 7 regions that are defined in the X–direction and 8 regions are used in the definition for the griding in the Y–direction. For each region the following information is needed

The number of cells of each region,
The distance to the end of the region, and
A grid clustering factor.

The nomenclature for each of these variables is given in the Q1 file. They are noted in Figure 1 of this report. In this figure the regions in both directions for CS61 are noted along with distance and clustering nomenclature. This input is used primarily for the description of lines and arcs in the data files for GGP. Figure 2 is the copy of a graphical display produced during the creation of the 2–D grid file. In this figure the full grid is displayed. Similarly plots for CS62 are supplied. In general, the data supplied for CS61 are used for CS62 through CS65. The dimensions of the various circles are used to calculate the corresponding squares in CS61. This is why some of the variables used to represent distance are set to 0.000000. A integer array is used as a marker to note the first region that contains an arc. The variable XGAP is the x–direction length of the gap over the orifice while IGAP is the number of cells in this gap.

Note in Figure 4 that it appears that lines overlap in the circular region. This is because some lines are overwritten with arc data. If this persists after a redraw in the GGP, major problems with the grid exist. More details in regard to the execution of GGP will be given later in this section.

When the initial grid is completed, the orthogonality of corner points of the circle can be improved (note Figure 5 and 6). This is done in the smoothing operations of the GGP. The number of cells affected by this is controlled by the variable ISOL located in the SATLIT file. In general these values will not need to be adjusted. Also plots of final grid CS63 and initial grid CS65 are shown in Figures 7 and 8.

The coding was designed so that major changes would be fairly straight forward. The input files has slots for 14 regions in each direction so that if more regions are needed in the future the accommodations can be made. Also, the number of cells for each region in the remaining types are not required but are obtained from the number of cells supplied for each region in TYPE 1.

TYPE 2 data produces two or three CS files. The first is for the exit of the engine which is also the same as for the inlet of the augmenter tube. The second is for the end of the augmenter lip. A third cross section may be required if the exit of the engine falls within the lip or within the sleeve. See Appendix A for more details. The only difference in these CS files will be the diameter of the two circles. Since the diameter of augmenter tube is larger than the engine exit additional cells are needed. The number of cells is controlled by variables NXAD and NYAD. For the case delivered two CS files (CS66 and CS67) were produced. The regions and initial grids for TYPE 2 are shown in Figure 9 through 11.

TYPE 3 data will produce three CS files. The first file is for the augmenter sleeve, the second is for the large diameter augmenter tube, and the last is for the small diameter augmenter tube. The difference in these files are due to the different diameters. The regions and grids (CS68, CS69, and CS70) are shown in Figure 12 through 16.

TYPE 4 data produces two CS files (CS71 and CS72). There is a small triangular room in front of the chimney. Constructing a grid from the whole room is impossible (grid lines would be on top of each other). The front half was removed. The rest was then included in the model. However due to orthogonality problems (see Figures 21 and 22) this room was blocked off and the ceiling was lowered. The first cross section represents the truncated front of the traingular room while the second represents the front face of the chimney. This is the first cross sections in which the first region does not start at a 0.0 X–coordinate value. A integer array element noted in the Q1 files takes this into account. A plot of regions and grids are shown in Figures 17 through 19.

The last grid is denoted by TYPE 5. It is located at the exit of the chimney. The input needed to produce this data file is taken from previously supplied information. The grid for CS73 is shown in Figure 20.

There is a integer array element that represents the stage of grid development. It is located in Group 6 of the Q1 file as is called IG (1). If the value of this element is set to 0, when the input files are executed, they will produce a set of data files for the GGP. If it is set to 1, then it will read the grid files produced by the GGP and create a 3–dimensional grid along with the other input files for the solver. If the grid is already created the value is set to 2 in order to bypass the grid creation coding.

In the form delivered, 13 data files for the GGP will be created during the first execution of the input files. At this time the user will then execute GGP as indicated in the documentation (probably done by entering runezm). The first item needed will be terminal type. Enter the appropriate value. Following this prompt, menus will appear on the screen. The following series of commands will go through these menus and produce a grid file.

PROMPT	ENTER	COMMENT
Model name	CS61	Use same name as file to be read in
EZ2 >	RE CS61	Reads in input file
EZ2 >	WR	Goes to menu to write grid
WRITE >	END	Writes grid
EZ2 >	END	End session

This is done when the grid to be produced is totally orthogonal (i.e. no circles). After the input file is read a redraw of the screen can be done through the REDR command. If lines cross after this point there is an error in the input file for the GGP. Looking at the grid may give clues as to the cause of the problem. If a grid needs to be smoothed (all files that contains a circle), the following commands will be needed.

PROMPT	ENTER	COMMENT
Model Name:	CS62	Use same name as file to be read in
EZ2 >	RE CS62	Reads in input file
EZ2 >	SM	Goes to smoothing menu
SMOOTH >	SO	Solves differential equations
SMOOTH >	REDR	Plots final grid
SMOOTH >	END	Returns to main menu
EZ2 >	WR	Goes to menu to write grid
WRITE >	END	Writes grid
EZ2 >	END	End session

After the creation of these 2–dimensional grid files, input in Q1 file is required for the formation of the final 3–dimensional grid. As in the specification of the grid in the X and Y–directions, the user must supply the number of regions, the distance to the end of the region, the number of cells, and the grid clustering factor. Allocations for 25 regions in the axial direction have been provided. As delivered, 20 have been specified.

The user must then supply the information for the building of the final grid. Four options are available 1.) Stack, 2.) Blend, 3.) Rotate, and 4.) End. Throughout the test cell the first two options are used to stack and blend the 2–dimensional grid files as needed, while the last two options create the grid in the chimney region. This information is passed to the SATLIT from the Q1 through an integer array.

2.2 Other Input

In group 9 of the input files most of the data for the physics of the model is supplied. These deal with flow rates, temperatures, mass fractions, etc. These are documented in the input files.

2.3 Relaxation

Relaxation is a numerical technique that allows the rate of change of various solved variables to be controlled. It is generally used to dampen the amount of change computed by the various computer codes. There are many views on the optimum settings of the relaxation parameters. In a problem of this size time constraints reduce the amount of effort in optimization of these parameters. The approach used was to reduce the relaxation (base values calculated on a cell residence time) at the start of a computational run and then apply tighter relaxation after a few hundred solution sweeps through the calculation domain.

The values of the relaxation parameters is given in the following table.

Table 1. Relaxation Parameters

Variable	Type	Initial Value	Final Value
P1	LINRLX	0.15	0.05
U1	FALSDT	0.001	0.0003
V1	FALSDT	0.001	0.0003
W1	FALSDT	0.001	0.0003
KE	LINRLX	0.10	0.10
EP	LINRLX	0.10	0.10
H1	FALSDT	0.005	0.001
C1	FALSDT	0.005	0.001

Note the two types of relaxations are discussed in the users guide. The final values were used after sweep 2758. (See following section for procedure to change relaxation.) It was observed during the reported run that monitor values downstream of the propeller tip were oscillating from sweep to sweep (i.e., for W1 values changed from 10 m/s to –5 m/s). This was stopped by clamping down on the pressure relaxation to 0.025 at sweep 948 and letting back up to a value of 0.125 at sweep 1103. During the first 900 sweeps of this computational run, the sources for the propeller had not been properly implimented. When the completed model was started from scratch it was noted that the pressure relaxation had to be lowered to a value of 0.10.

2.4 Other Controls

Depending on computer systems, it may take a few weeks to obtain a fully converged solution. The code allows for restarts using previous data. For some cases this may not be the best procedure as compared to one long run. Because of this various controls were put in the GROUND coding that allows the user to vary items during one long run. This coding allows the user to:

1. Abort a run with standard output produced,
2. Modify pressure relaxation,
3. Modify turbulence relaxation,
4. Modify velocity relaxation,
5. Modify scalar relaxation,
6. Dump a restart file on demand,
7. Change frequency of monitor printout,
8. Change frequency of residual printout,
9. Change the number of variables in the monitoring values printed, and
10. Change two moni.or locations.

This is accomplished by:

1. Providing a file called ABORT,
2. Providing a value in the F12.8 Format in a file called RELAXP,
3. Providing two values in the 2F12.8 Format in a file called RELAXT,
4. Providing three values in the 3F12.8 Format in a file called RELAXV,
5. Providing two values in the 2F12.8 Format in a file called RELAXS,
6. Providing a file called DUMPIT,
7. Providing a value in the I5 Format in a file called TSTMOD,
8. Providing a value in the I5 Format in a file called NPRMOD,
9. Providing four values in the 4I2 Format in a file called IGGMOD (value of 1 activates printout while a value of 0 deactivates), and
10. Provide three values in the 3I3 Format in a file called ML2MOD or ML3MOD (values are for the IX, IY, and IZ locations).

2.5 Additional Printout

In addition to the standard output the following printout is provided:

1. Ten monitoring locations,
2. The maximum and minimum values for certain variables,
3. Convergence information,
4. Pumping ratios, and
5. Heat transfer information.

Note the previous section provided some information about control of the monitoring printout. The max–min printout may give clues to problem areas. Monitoring printout can then be shifted to these locations. The convergence information gives a mass and momentum error based on mass and momentum sources. A value of under 1% for mass and 3% for momentum should be acceptable. In addition, the pumping ratio for the engine is printed. When these value become asymptotic, this may indicate convergence. Printout is also provided for the heat transfer through the augmenter tube in the building and in the chimney. Similarly asymptotic values point toward convergence.

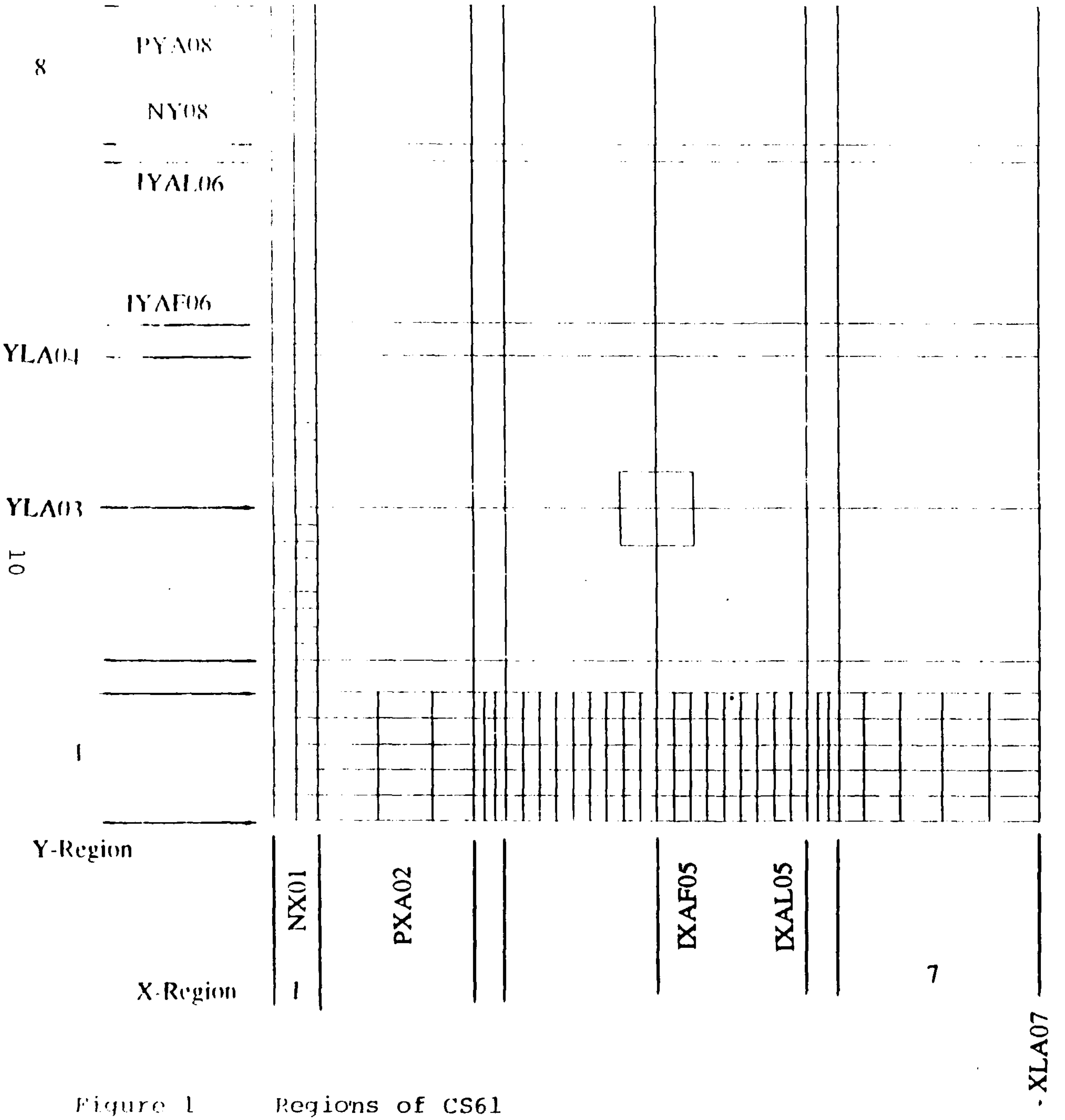


Figure 1   Regions of CS61

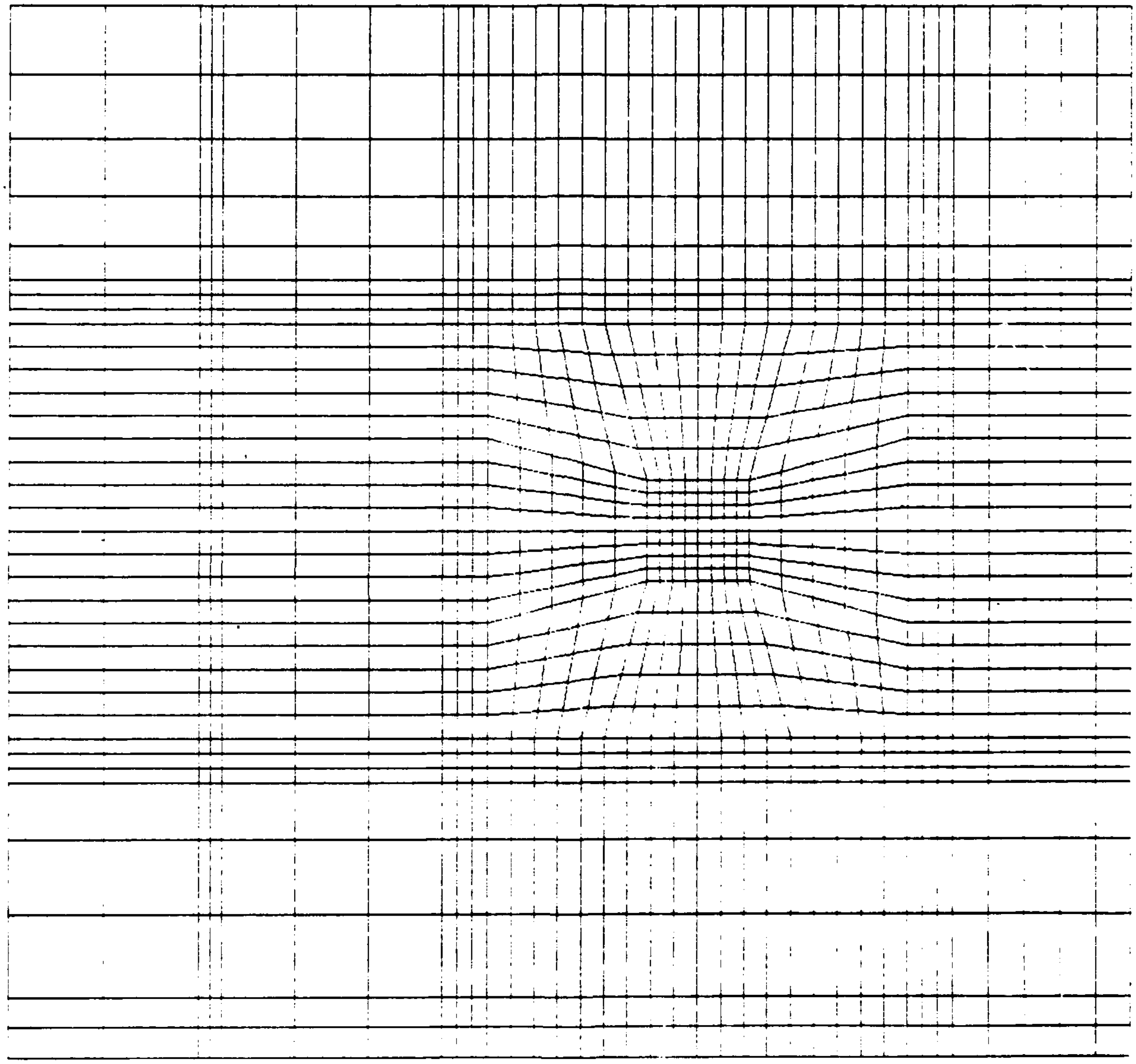


Figure 2 Grid of CS61

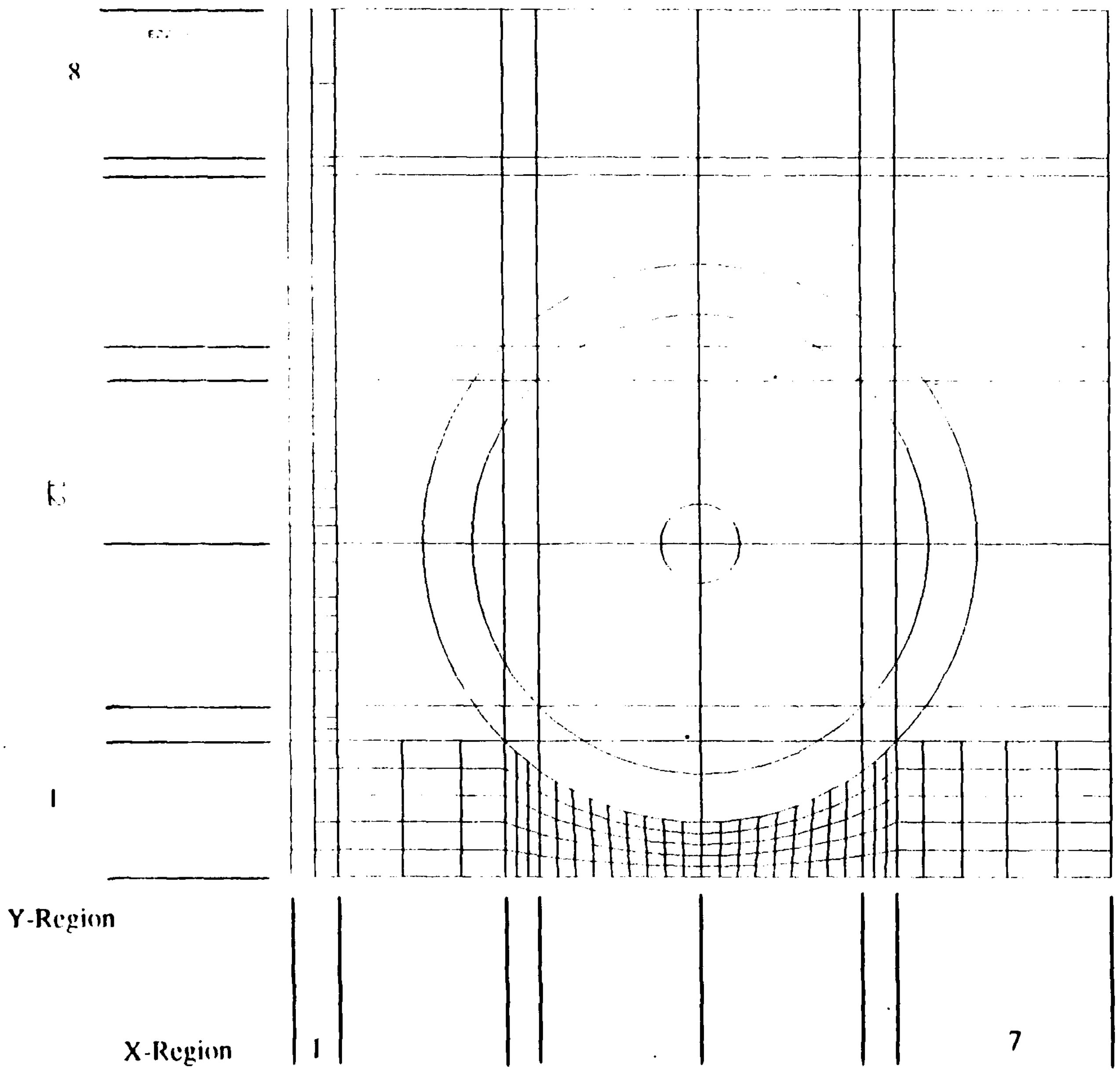


Figure 3 Regions of CS62

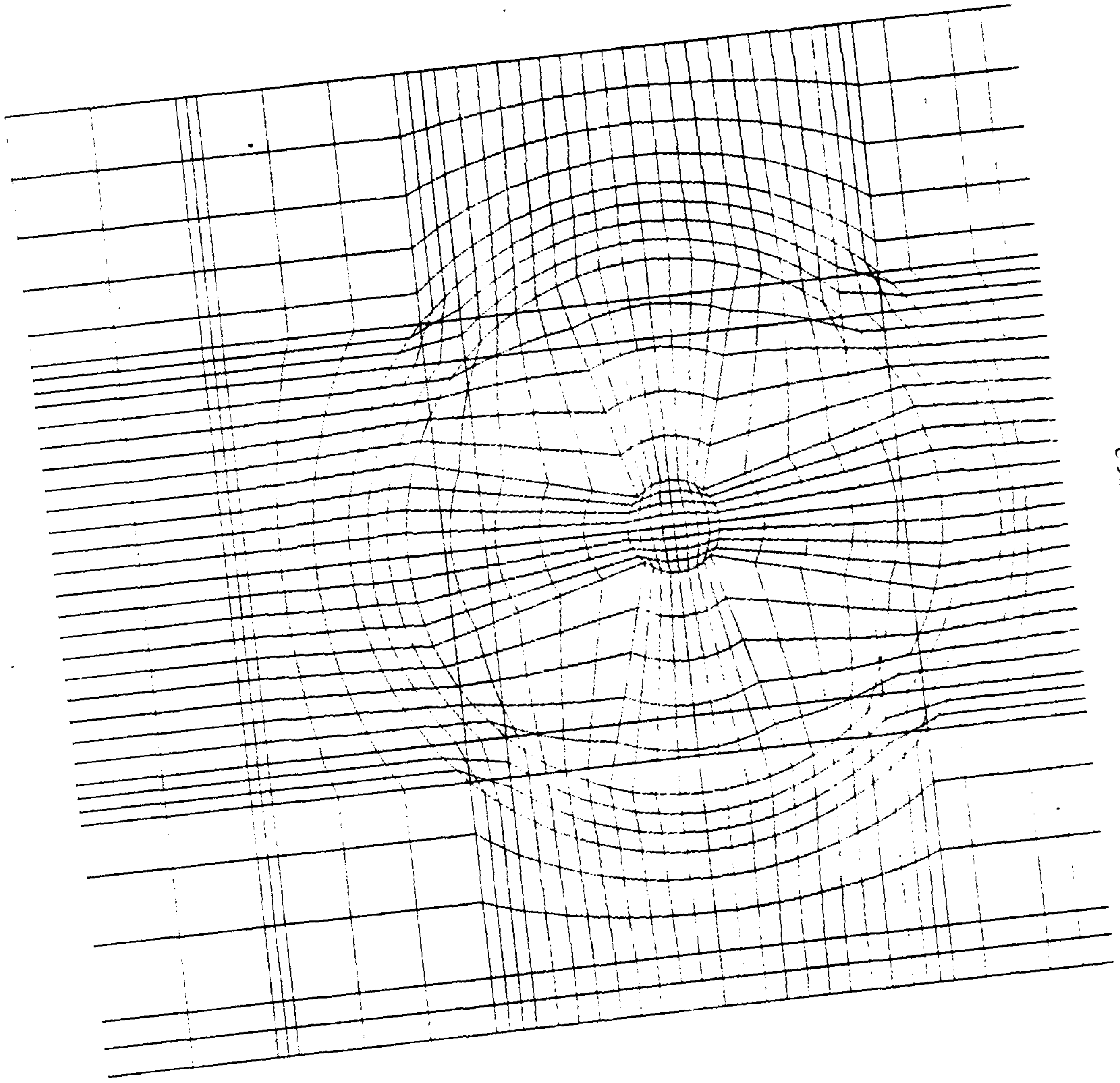


Figure 4 Initial Grid of CS62

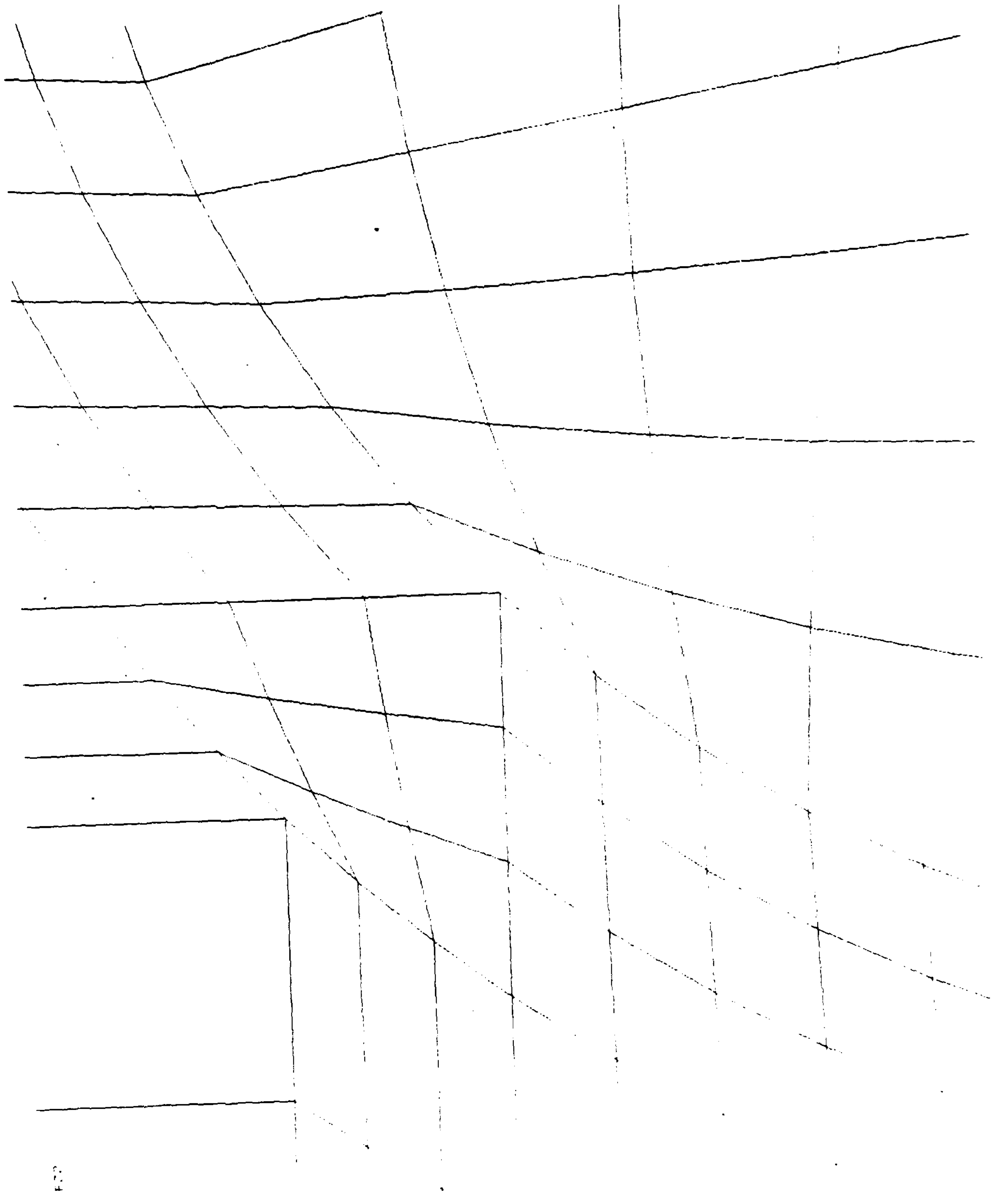


Figure 5. Enlargement of Initial Grid of CS62

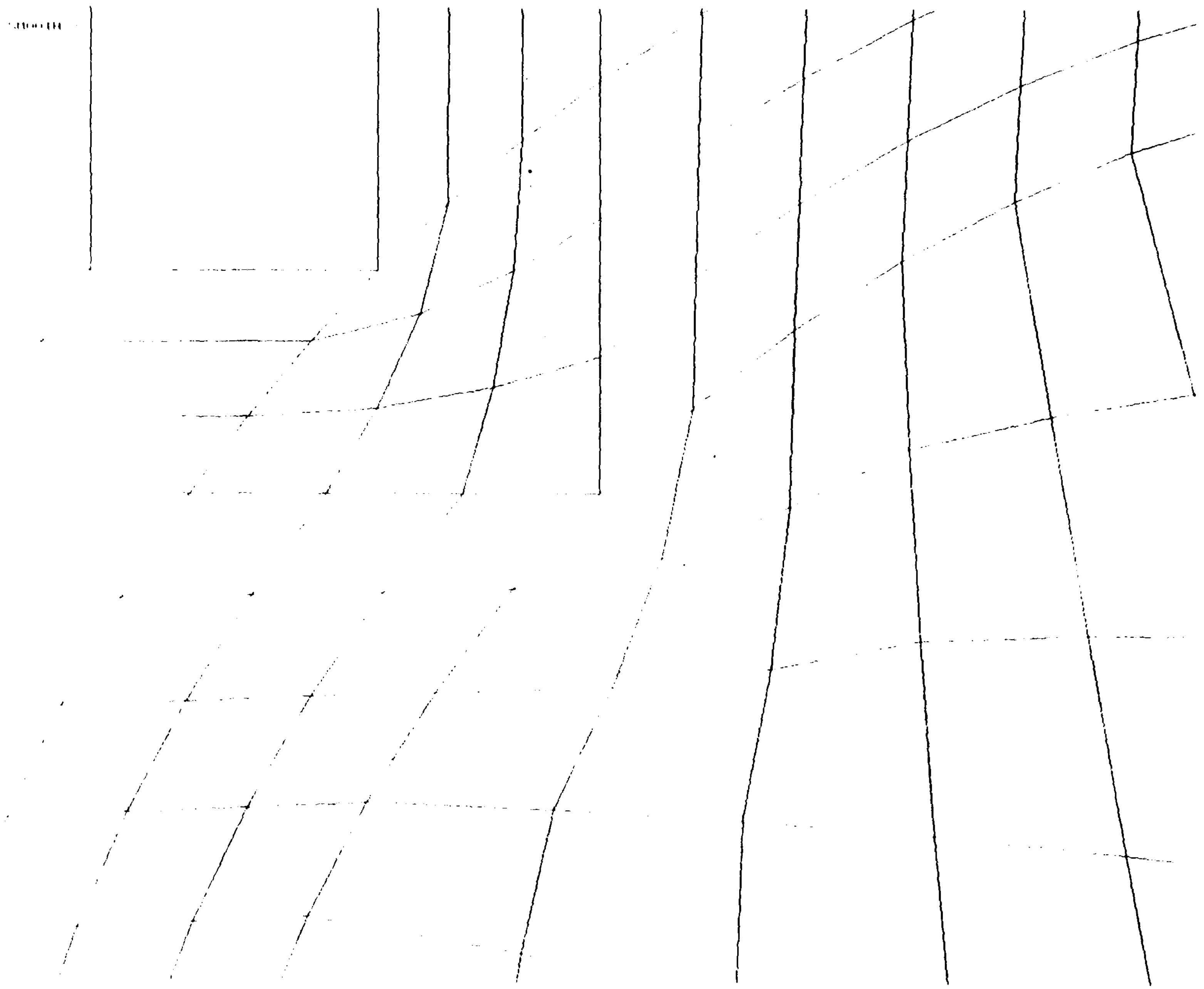


Figure 6 Enlargement of Final Grid of CS62

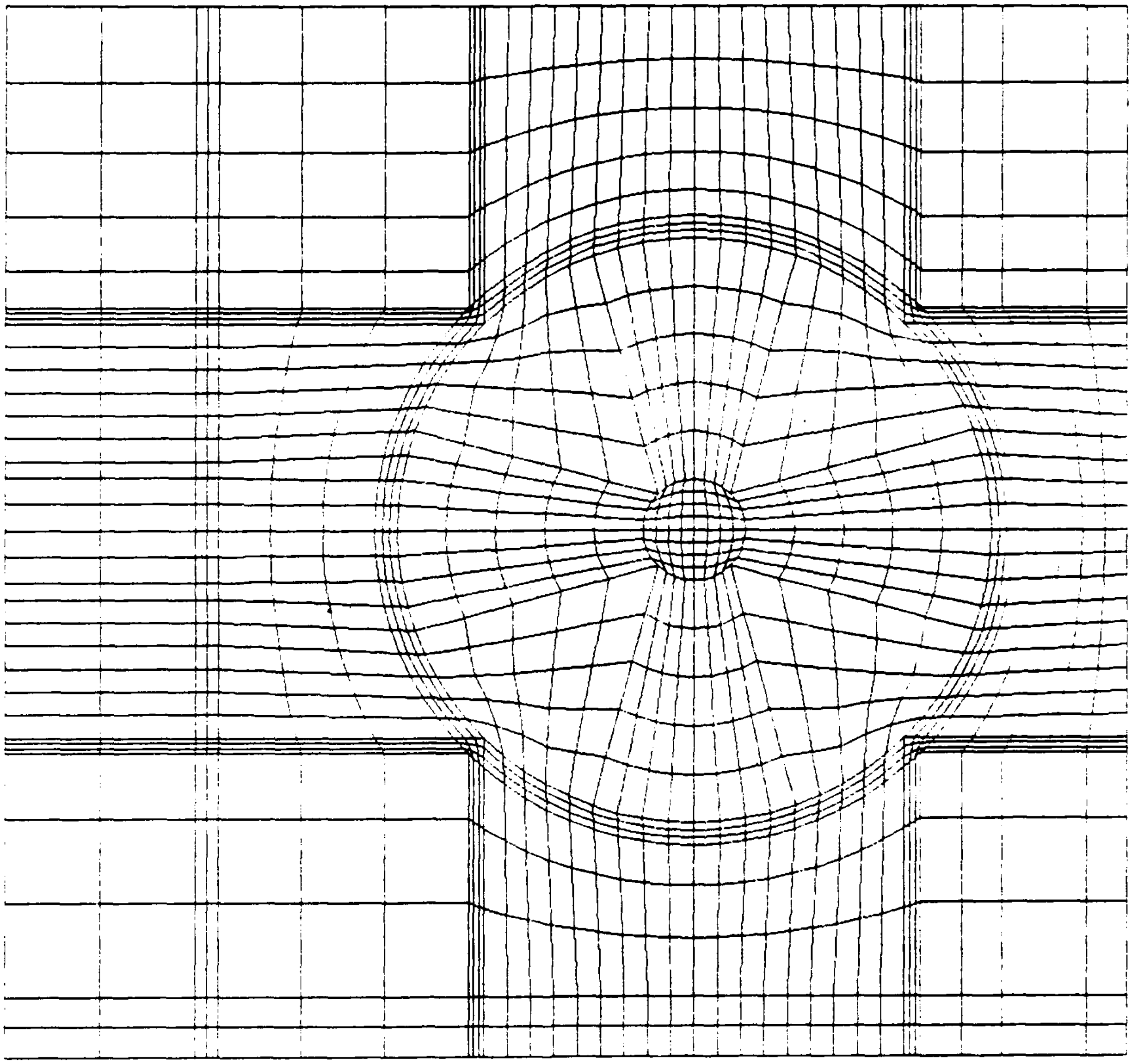


Figure 7 Grid of CS63

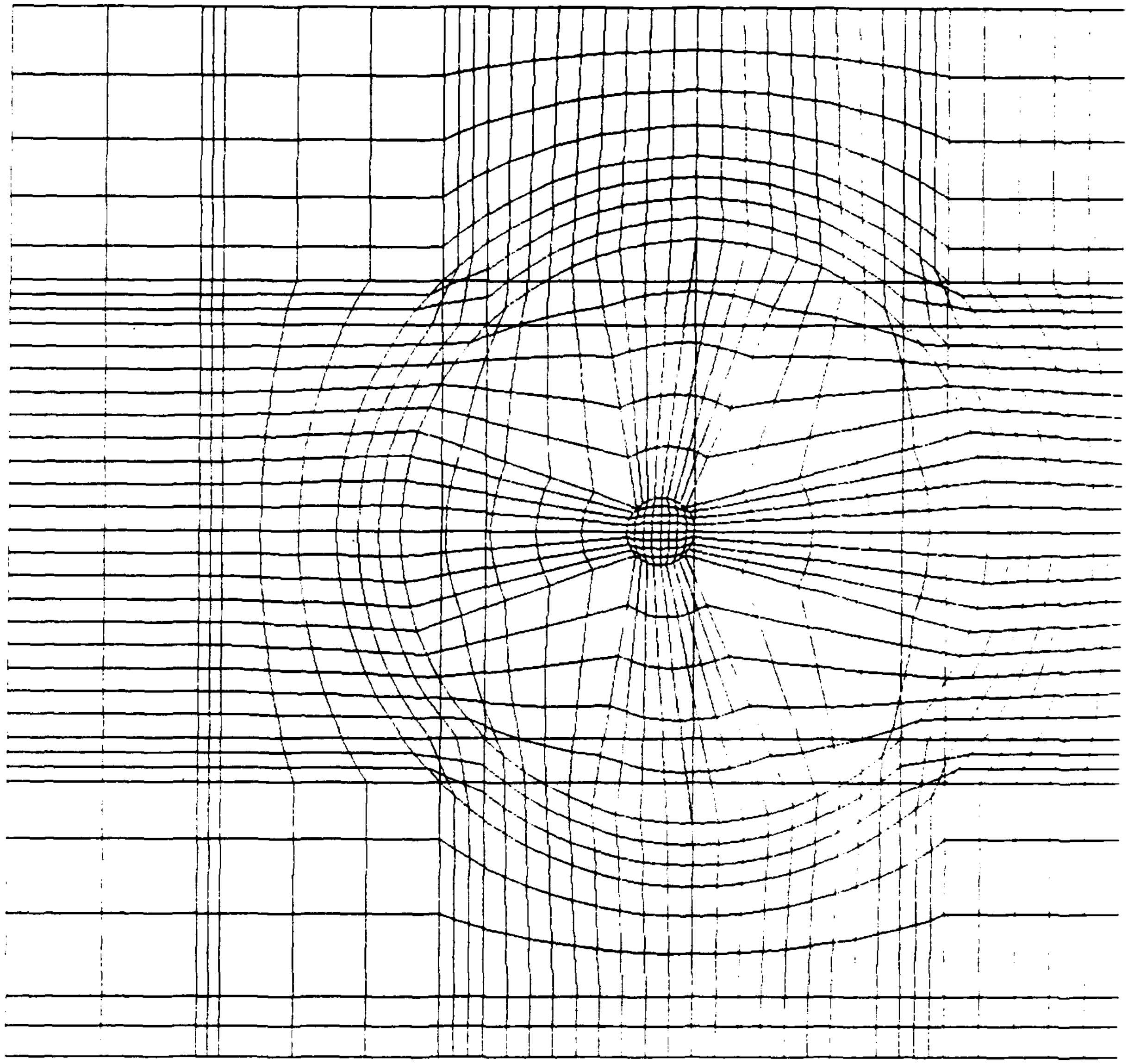


Figure 8 Initial Grid of CS65

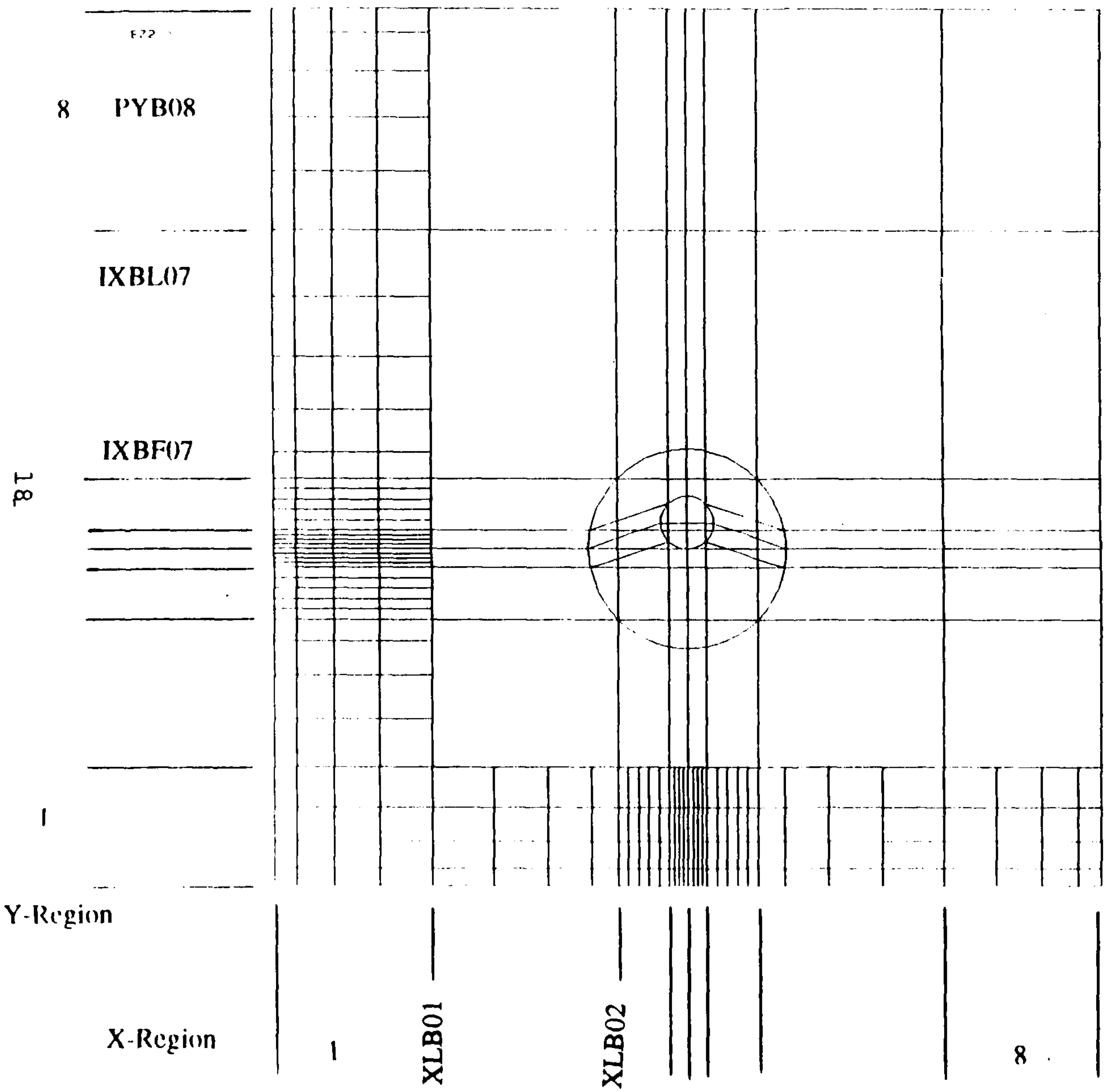


Figure 9 Regions of CS66

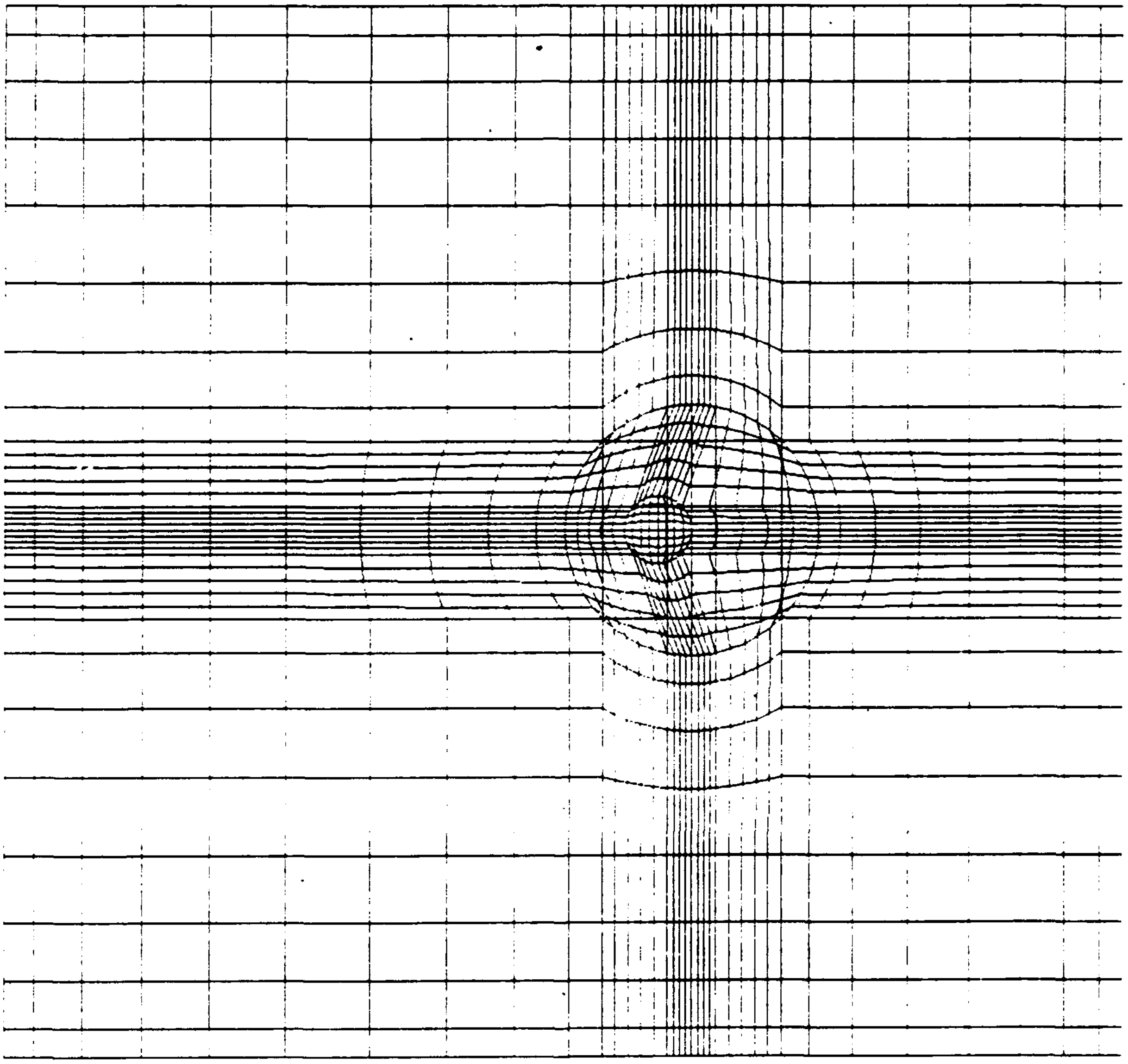


Figure 10 Initial Grid of C366

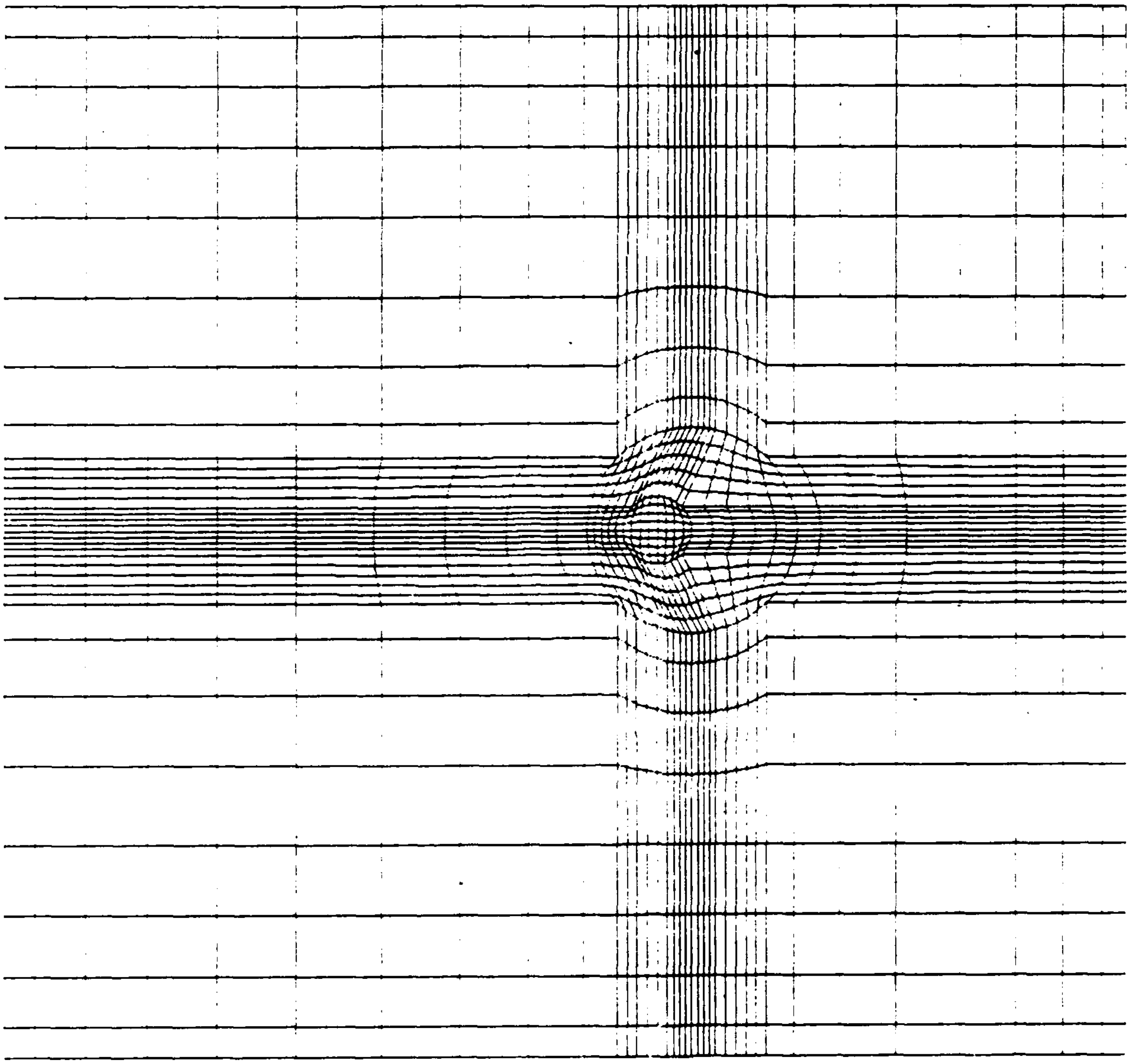


Figure 11 Final Grid of CS67

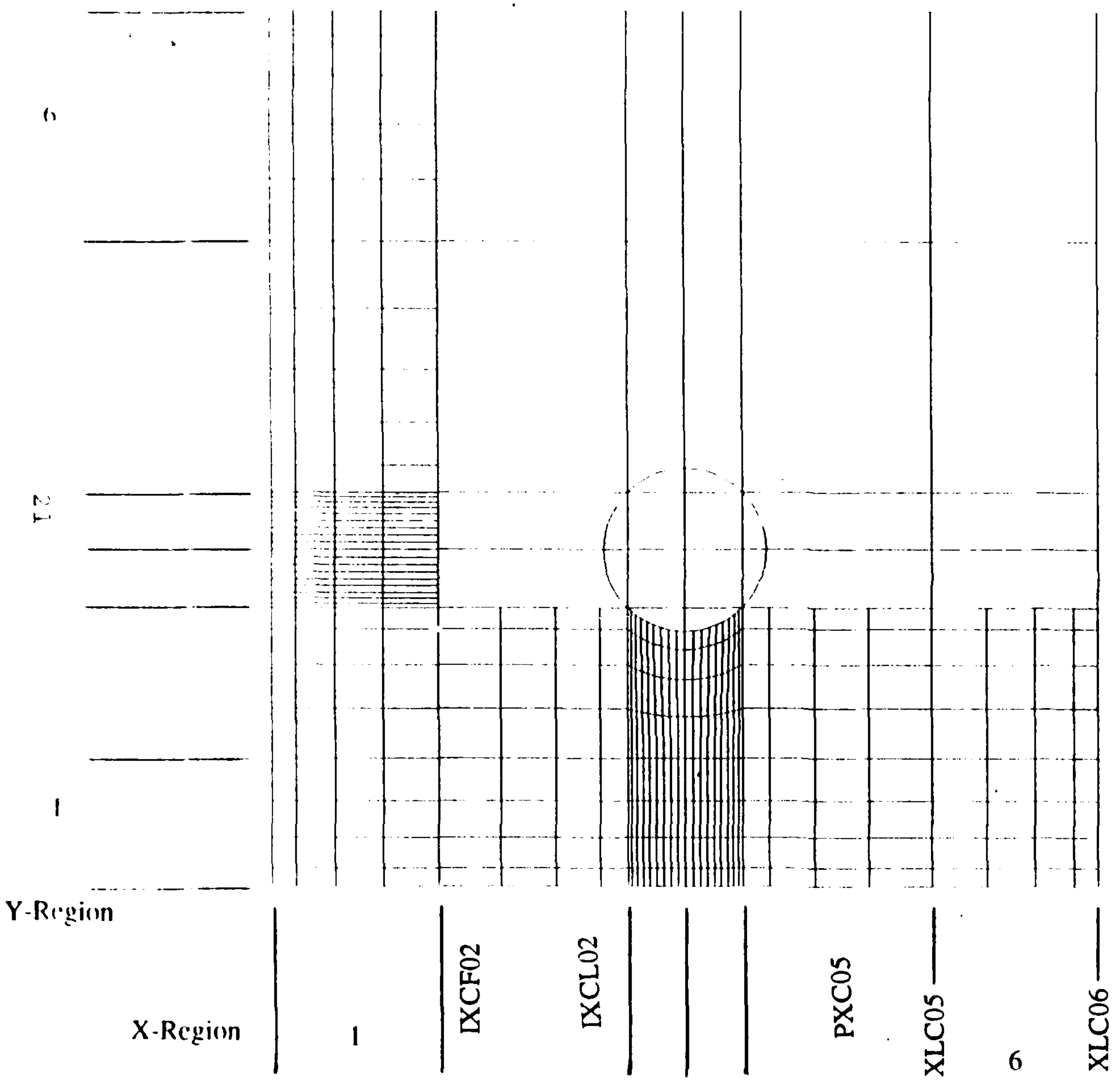


Figure 12 Regions of CS68

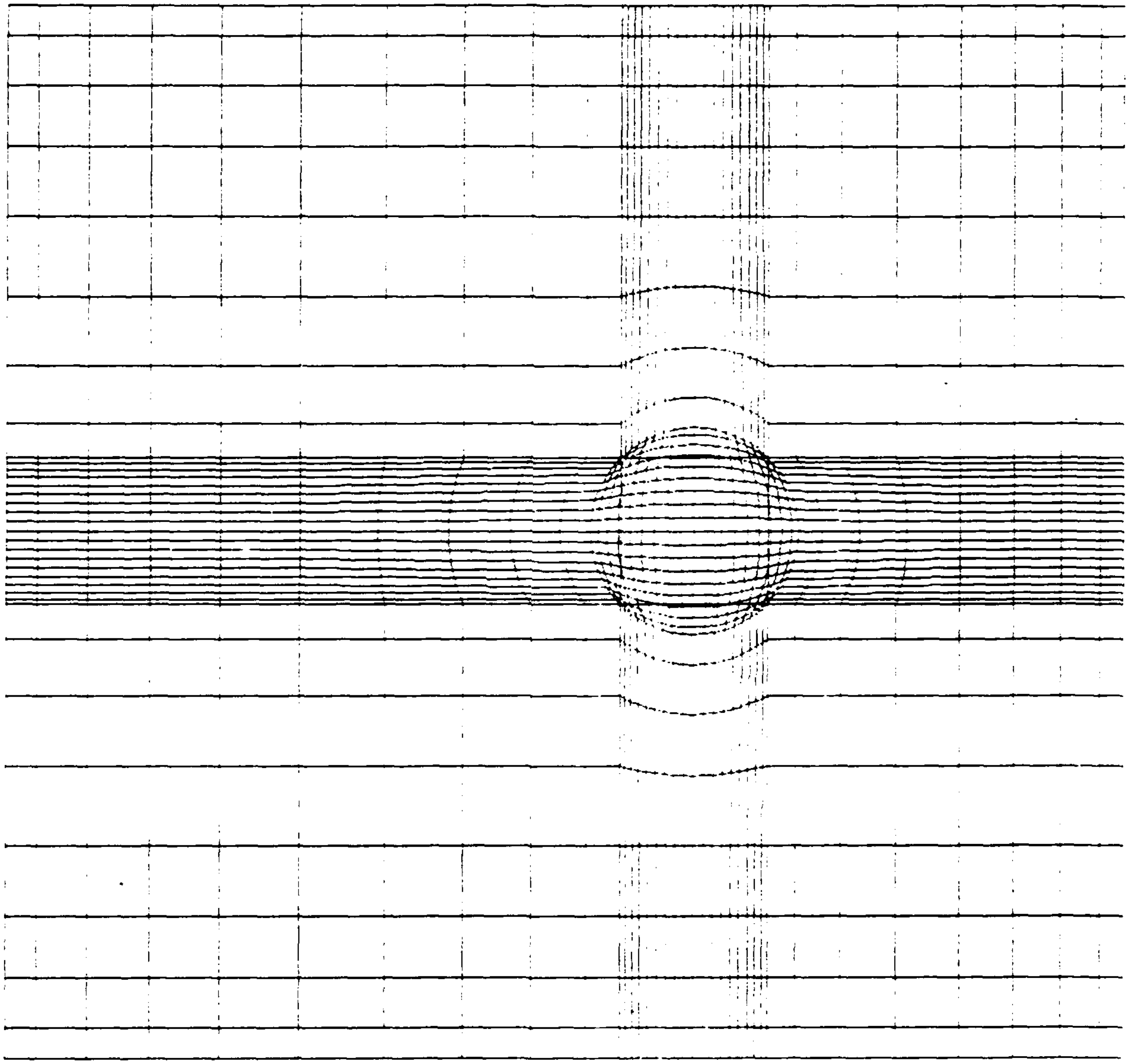


Figure 13 Initial Grid of CS68

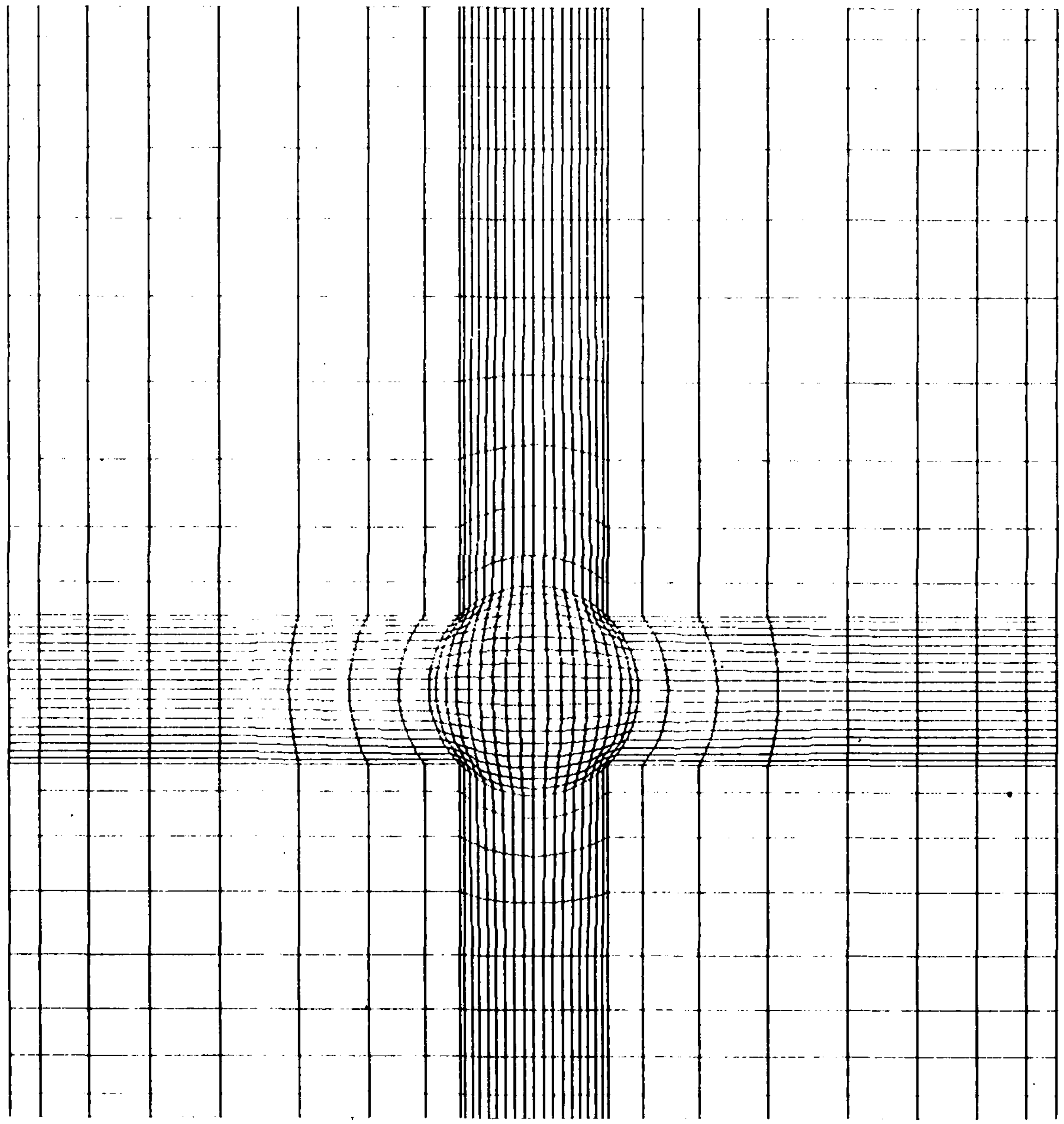


Figure 14   Final Grid of CS69

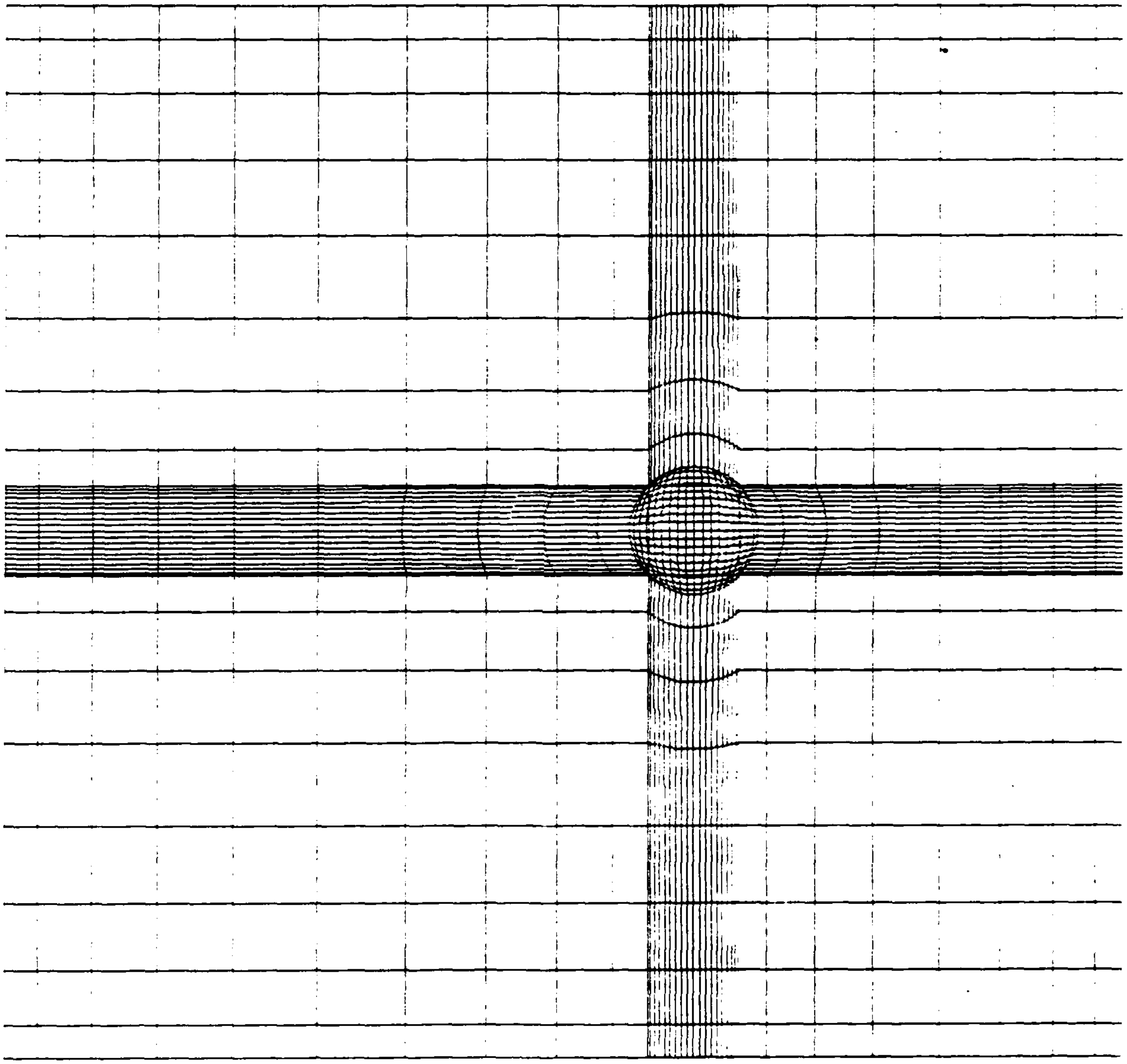


Figure 15 Initial Grid of CS70

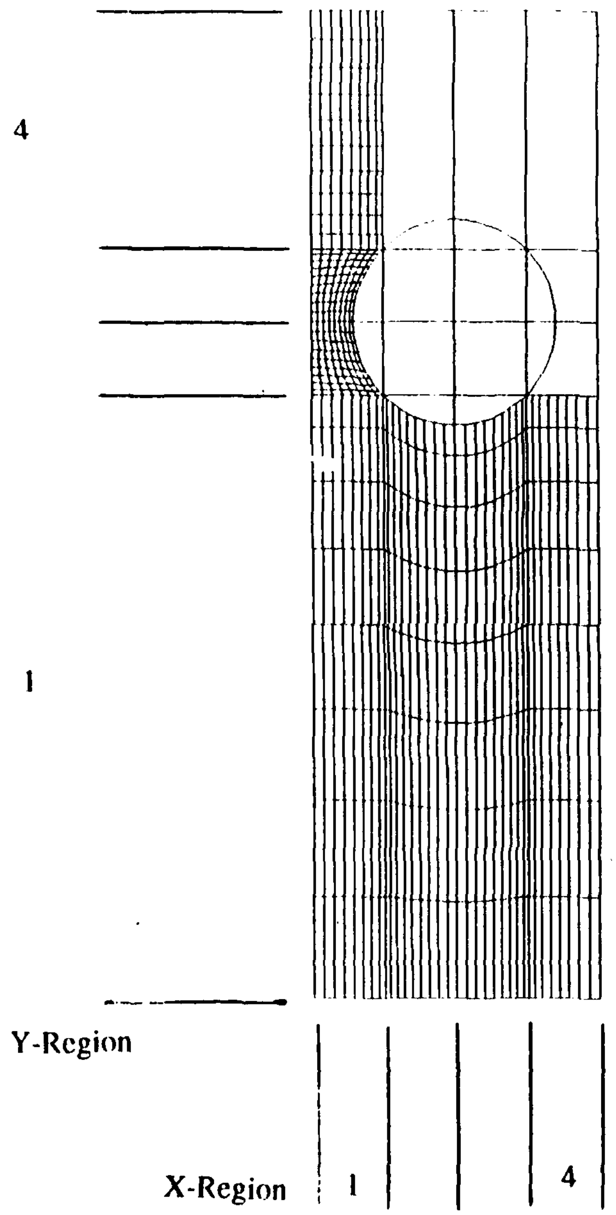


Figure 16 Regions of CS71

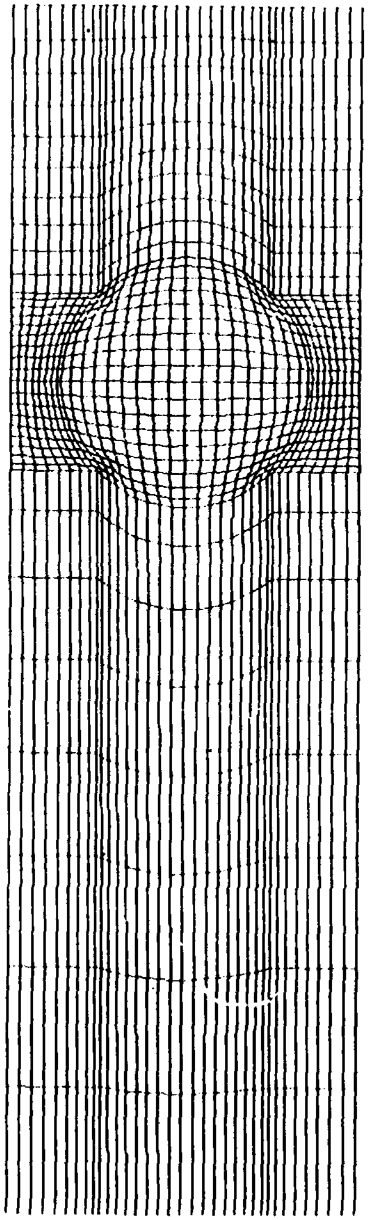


Figure 17 Final Grid CS71

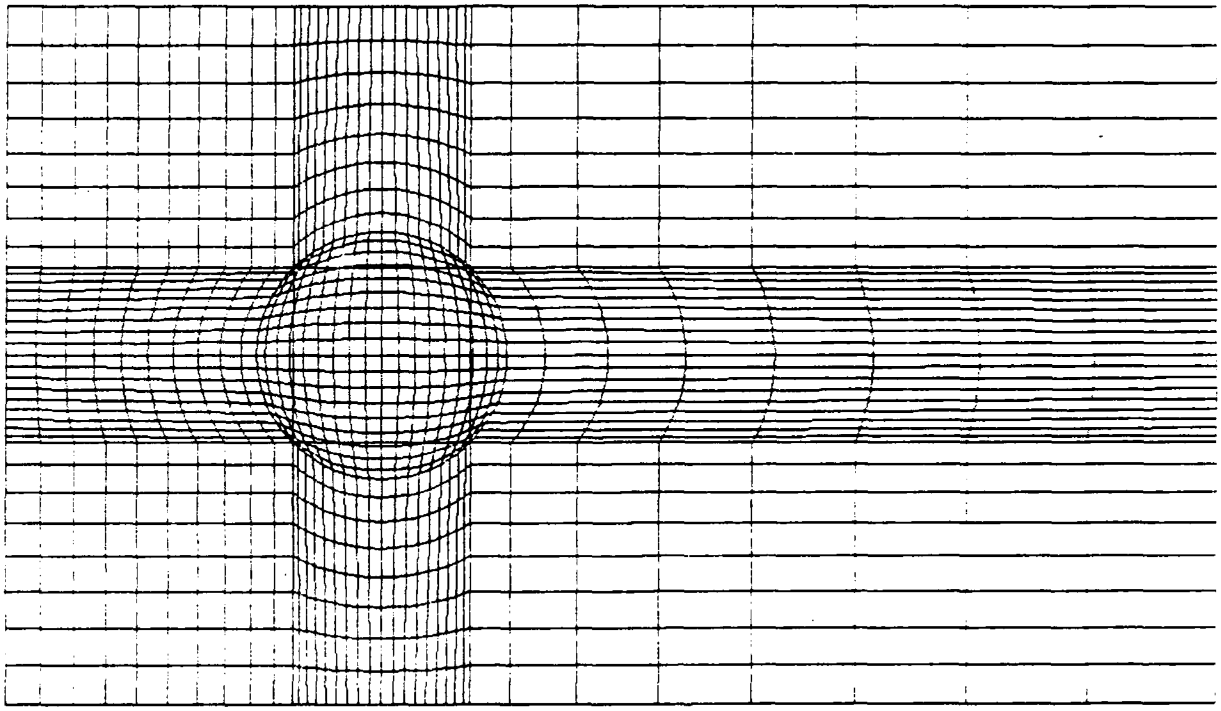


Figure 18 Initial Grid CS72

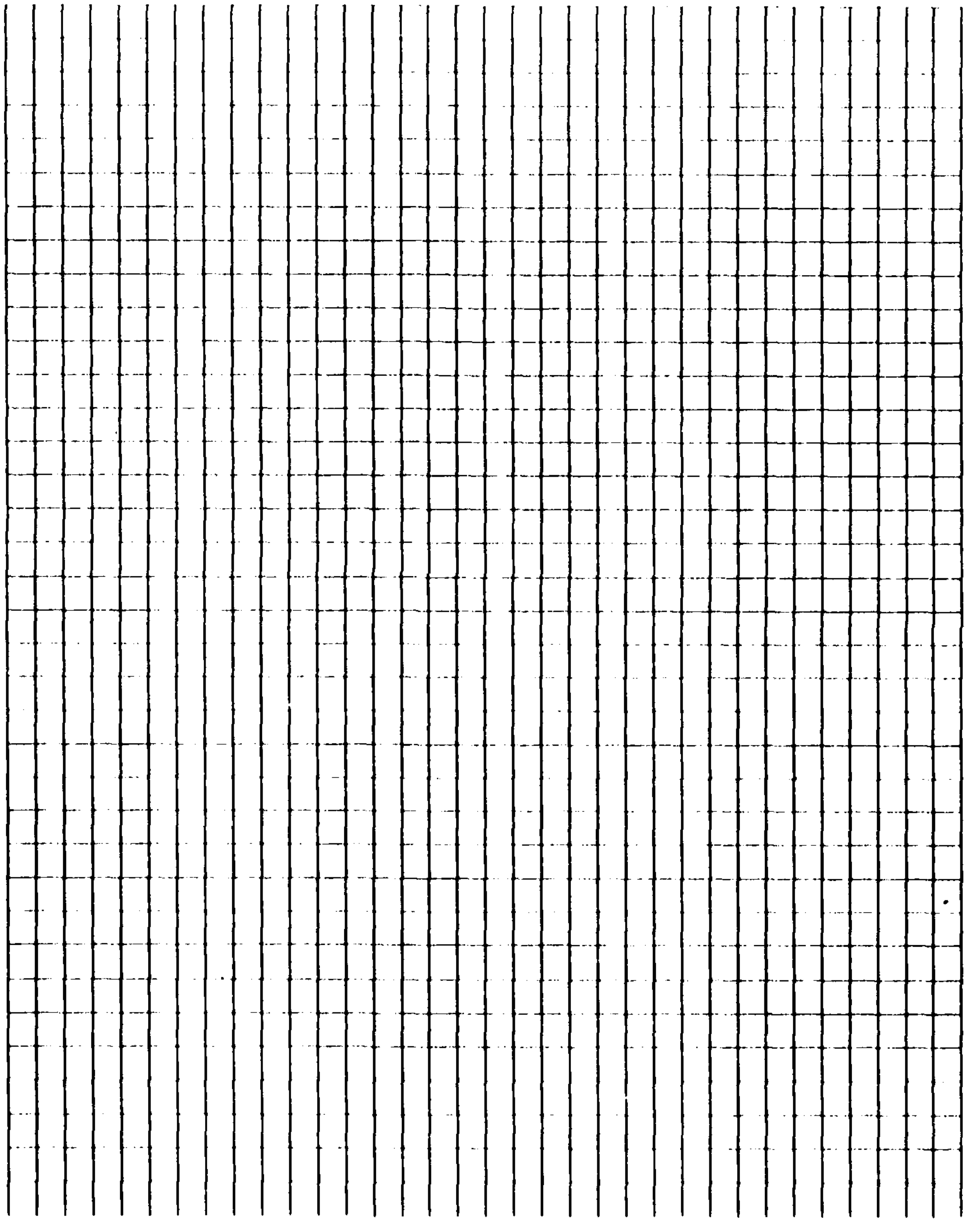
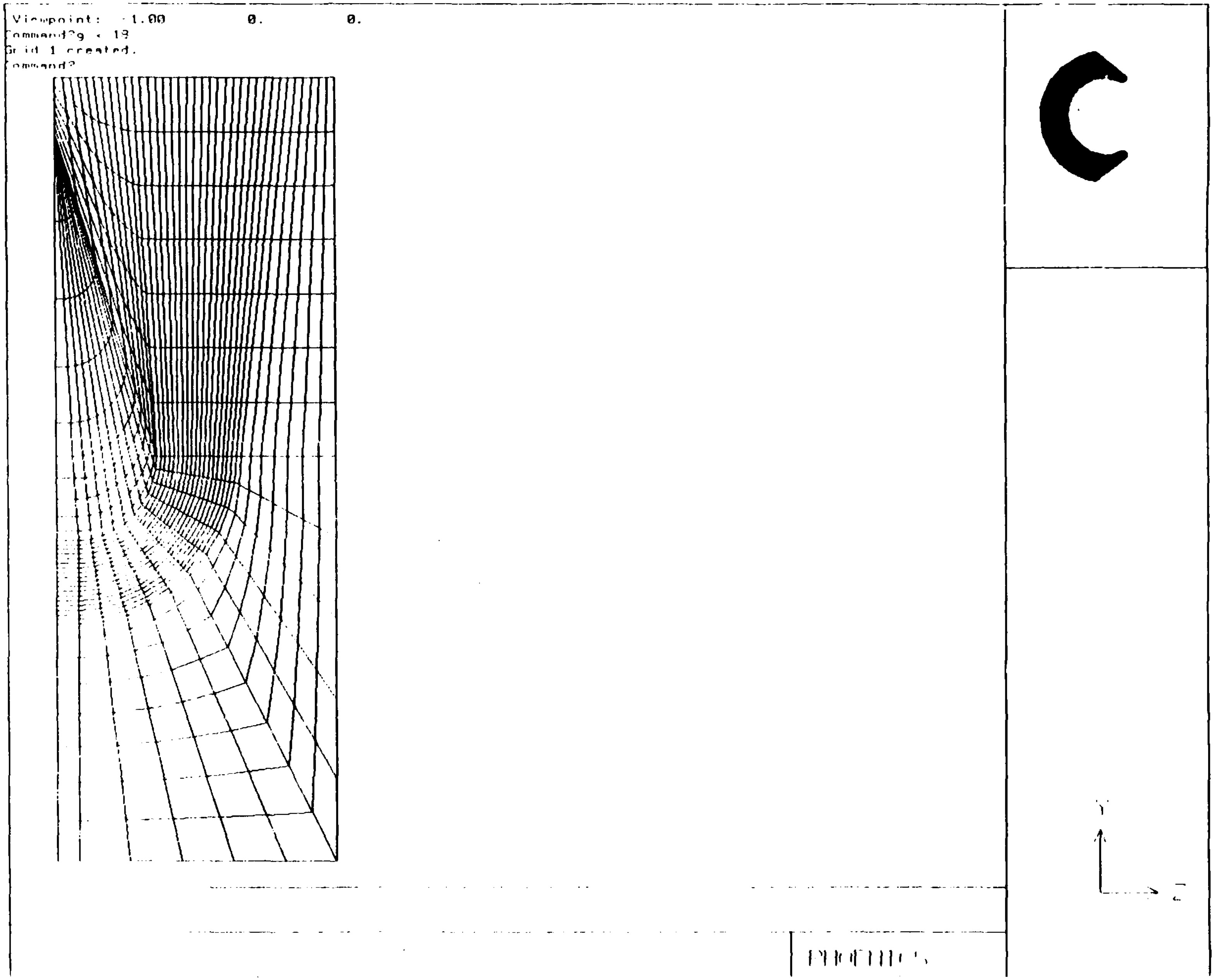


Figure 19   Grid of CS73



```
Viewpoint:  -1.00         0.         0.
Command?g < 19
Grid 1 created.
Command?
```

Figure 20   Grid of Original Chimney Section

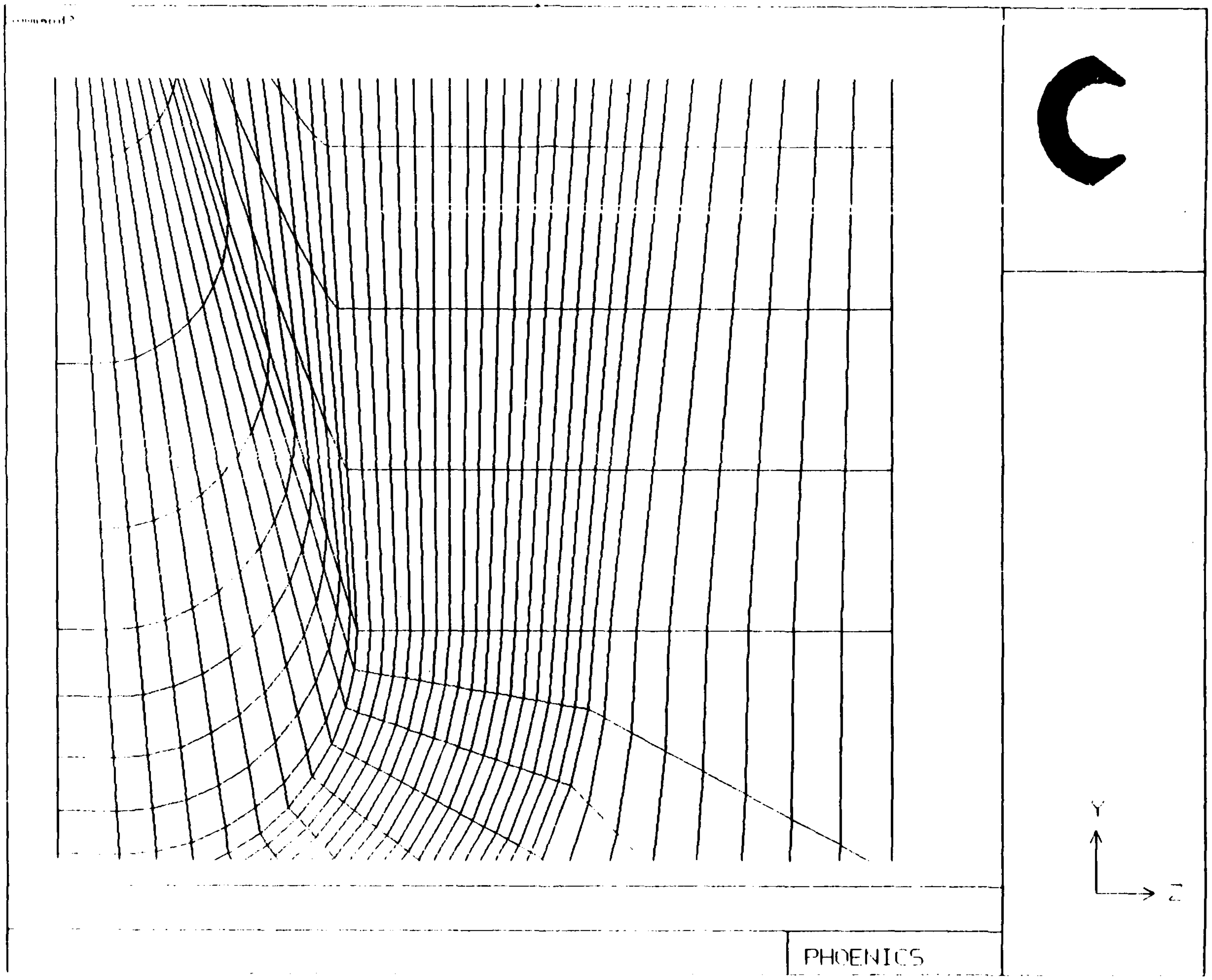
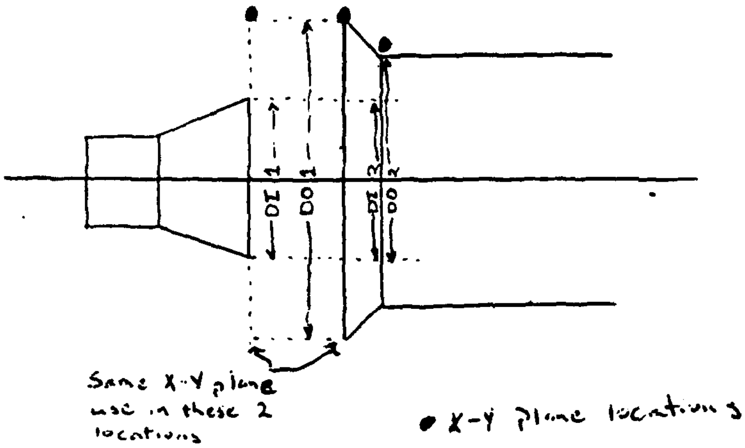


Figure 21 Enlargement of Original Grid of Chimney Section

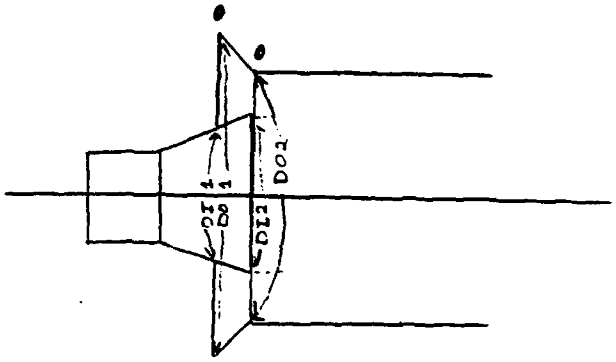
APPENDIX A

LOCATIONS OF DI_x + DO_x

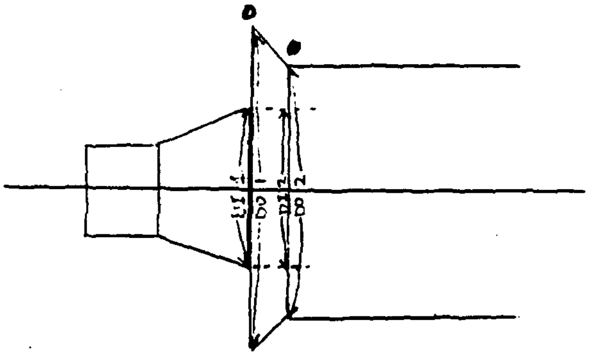
TYPE A



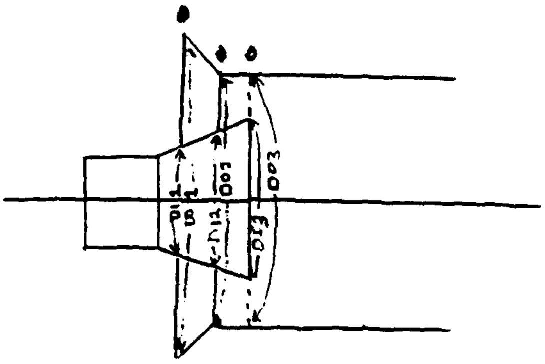
TYPE D



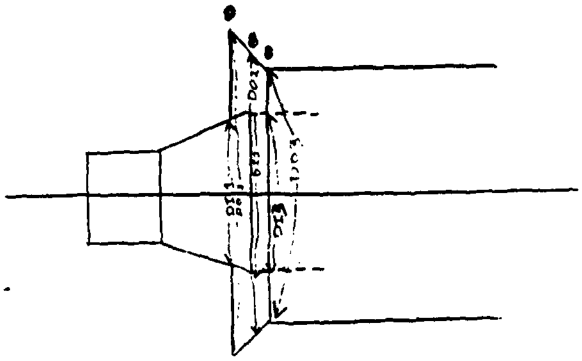
TYPE B



TYPE E



TYPE C



D → Diameter

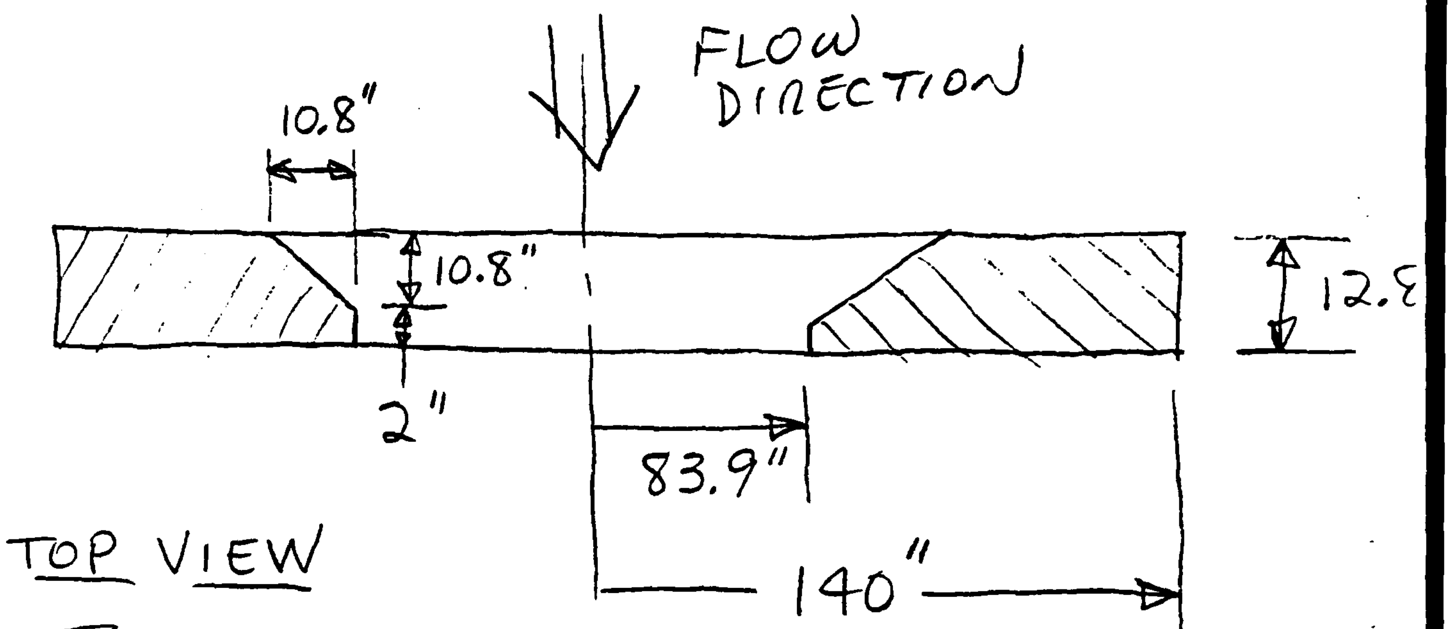
I → Inner

O → Outer

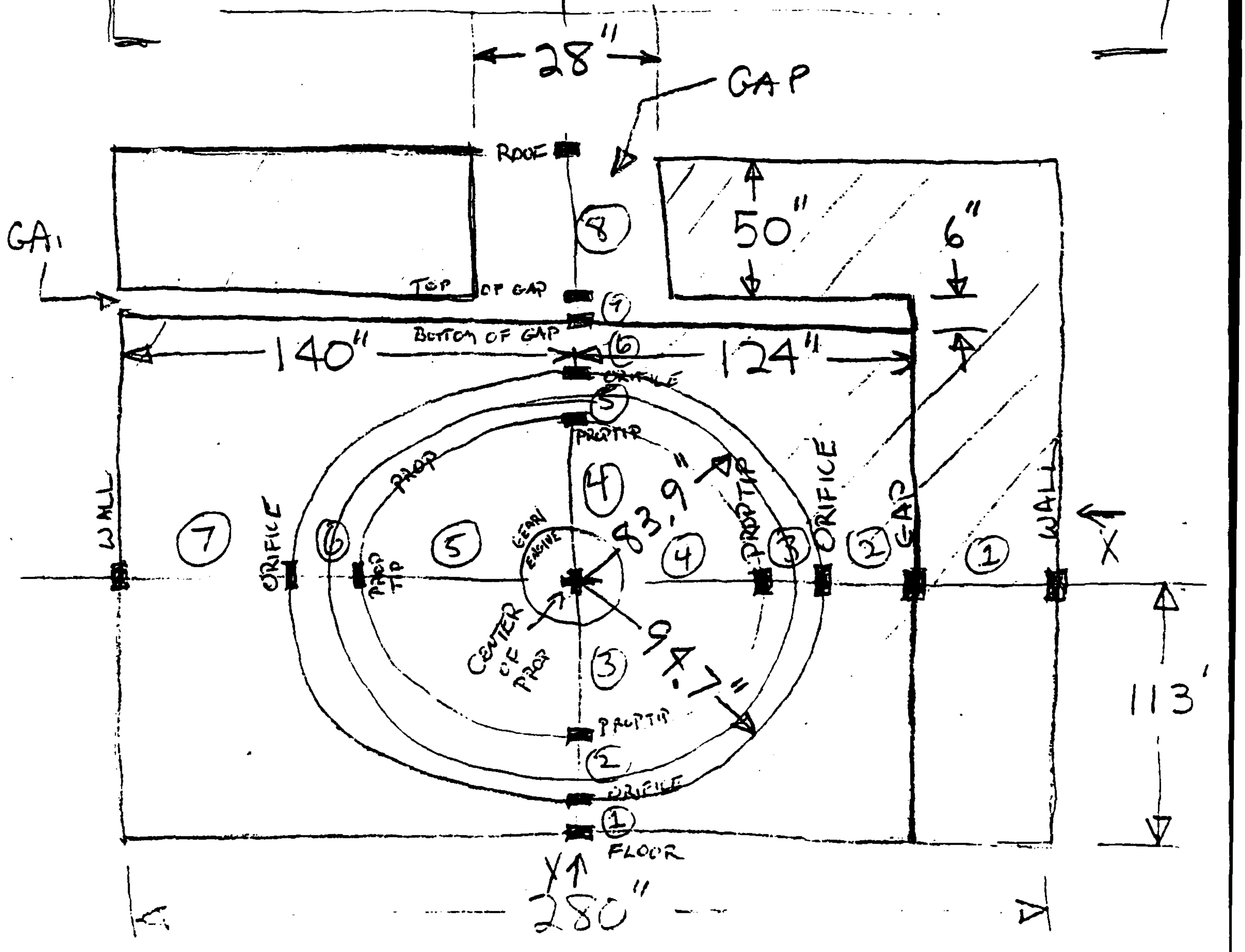
1 → Plane 1

2 → Plane 2

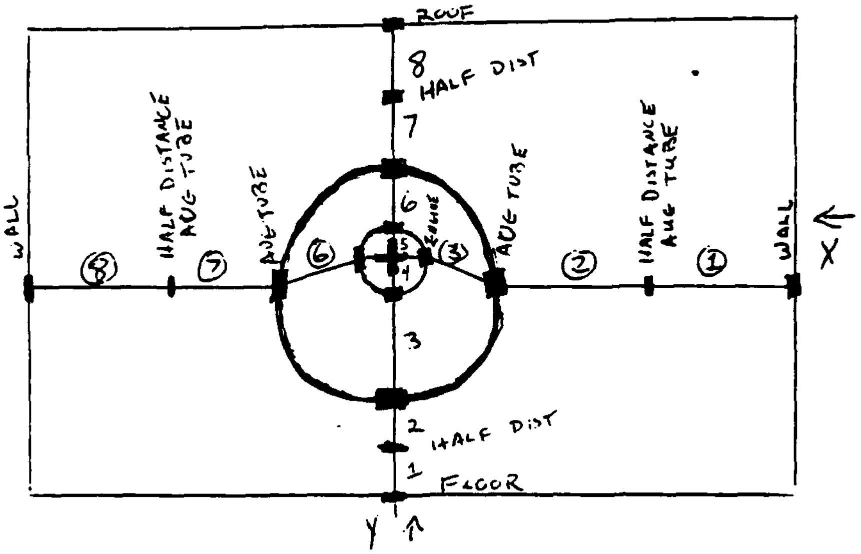
3 → Plane 3



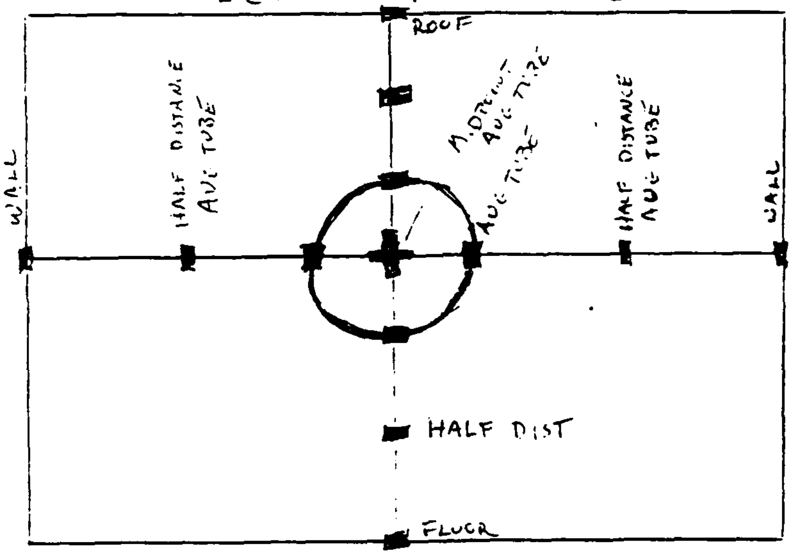
LOCATIONS OF REGIONS FOR TYPE 1



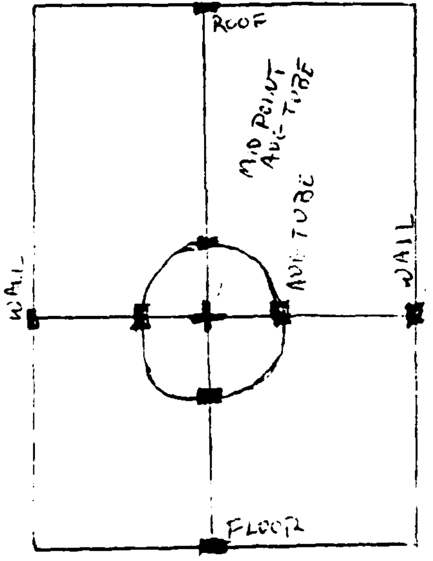
REGIONS FOR TYPE 2



REGIONS FOR TYPE 3

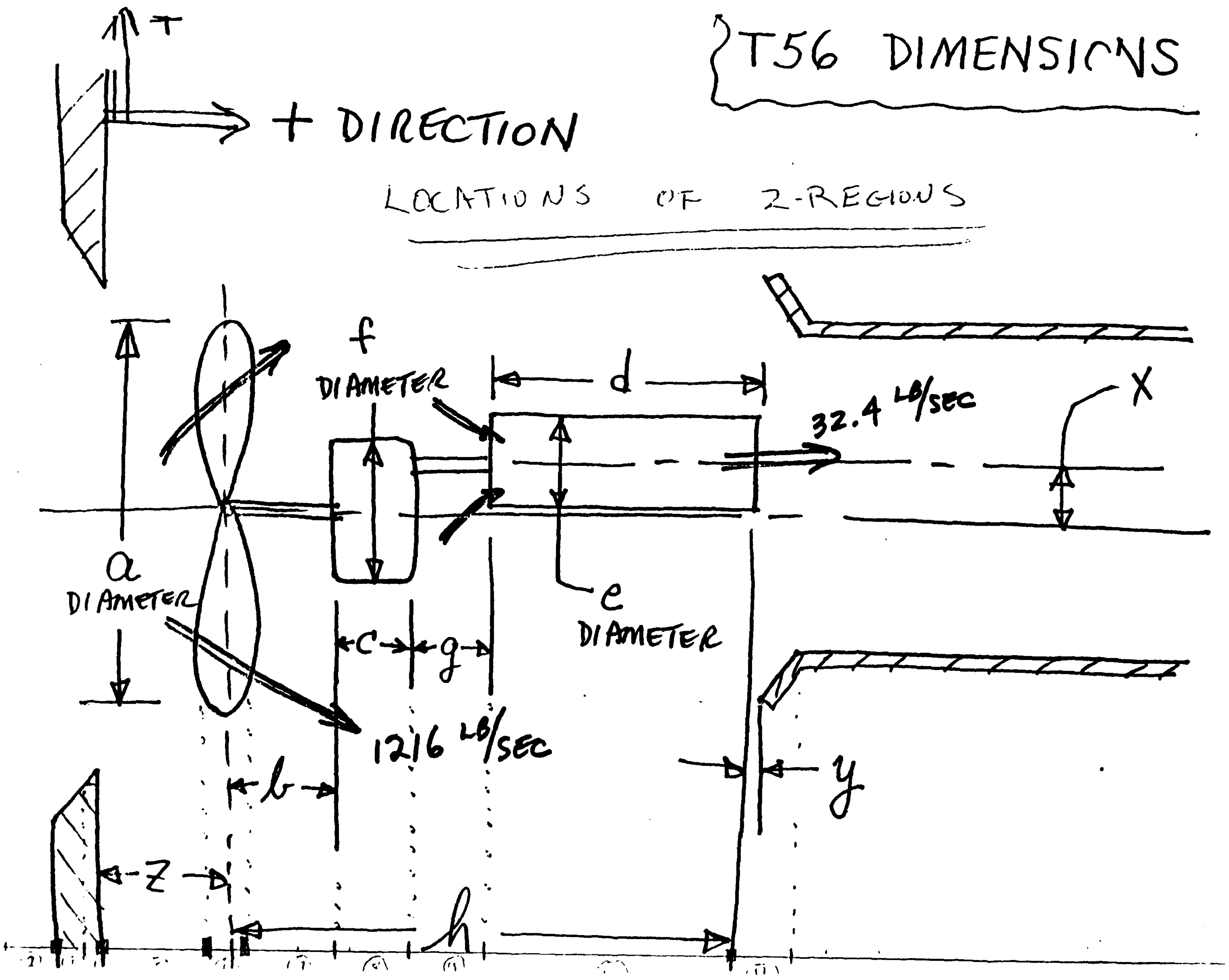


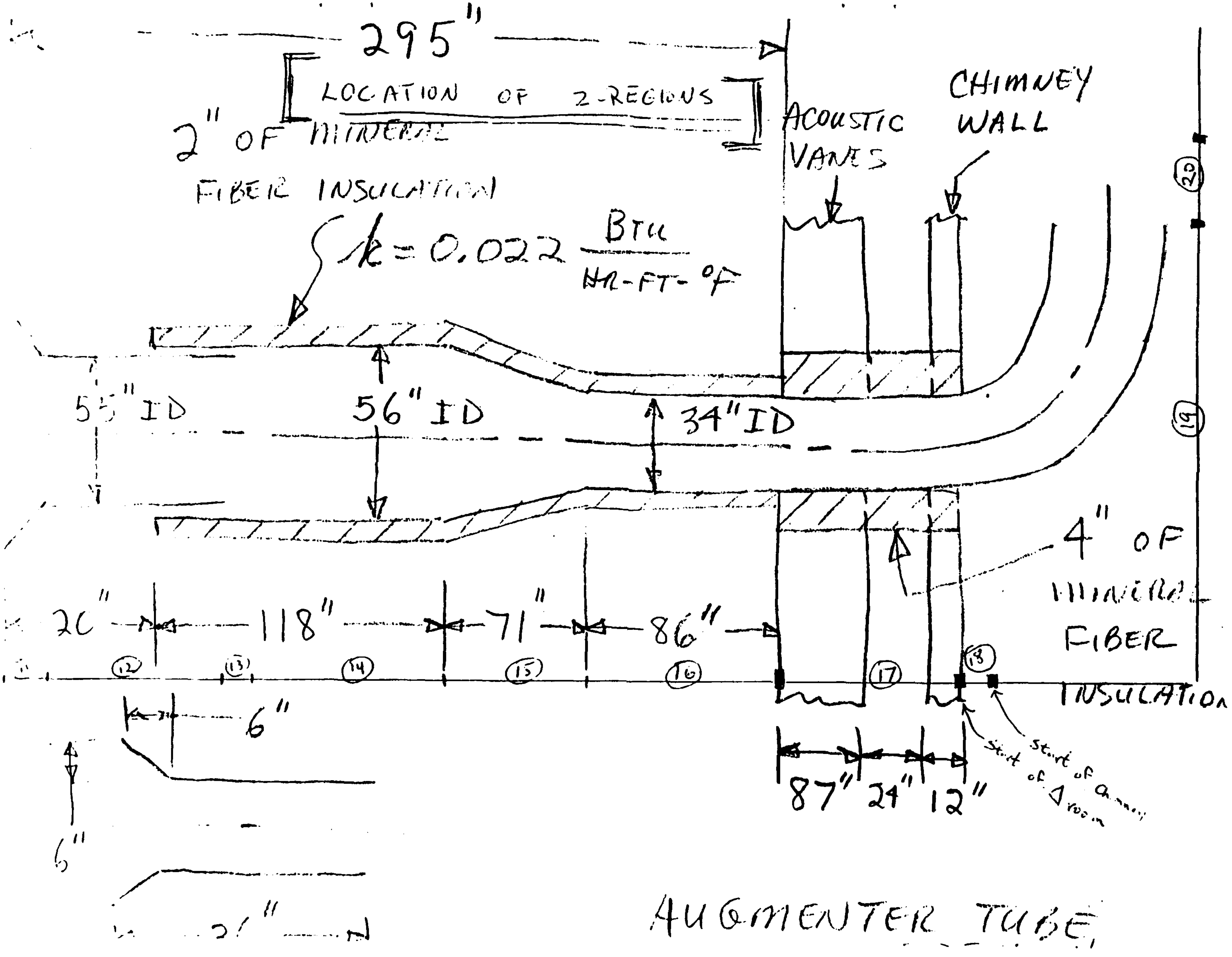
REGIONS FOR TYPE 4



T56 DIMENSIONS

LOCATIONS OF 2-REGIONS





AUGMENTER TUBE

APPENDIX B

```
TALK=F;RUN(1,1)
   GROUP 1. Run title and other preliminaries
TEXT(NCEL:  TEST CELL--TURBOPROP ENGINE)
 ***********************************************************************
 *********                   GRID SECTION                  ************
 ***********************************************************************
 ***   PRELIMINARY: Grid generation is an art form.  This           ***
 ***     model attempts to make this process as painless as         ***
 ***     possible.  Several assumptions will be made during         ***
 ***     this procedure.  Each will be stated at an appropriate     ***
 ***     time.  These assumptions will limit the parametric         ***
 ***     geometrical studies that can be accomplished.              ***
 ***     Geometric changes as called for by the contract will       ***
 ***     be possible and fairly easy to implement.  This method     ***
 ***     will not make it easy for radical modifications to be      ***
 ***     modeled.  However, with the appropriate assistance such    ***
 ***     changes should be possible.                                ***
 ***********************************************************************
 *
 *
 ***********************************************************************
 ***                                                                ***
 ***   PREMISE:  The grid for the test cell is created from a       ***
 ***     2-dimensional grid generation package.  This package       ***
 ***     produces several X-Y cross sections.  These sections       ***
 ***     are then stacked, blended, or rotated to produce the       ***
 ***     entire computational domain.  In order to do this, grid    ***
 ***     information data is supplied by the user in the Q1         ***
 ***     file.  This information is then transferred to SATLIT      ***
 ***     where the input files for the grid generation are          ***
 ***     created.  The user must then manually run the grid         ***
 ***     generation program to produce a plane of X-Y grid          ***
 ***     points for each input file.  After this the user will      ***
 ***     then rerun the preprocessor (Q1-SATLIT) at which time      ***
 ***     the full computational grid will be produced.  This is     ***
 ***     controlled by the setting of IG(1) in GROUP 6.  If IG(1)   ***
 ***     is set to 0 the execution of Q1-SATLIT produces the        ***
 ***     input files for the grid generation package; if set to 1   ***
 ***     execution of Q1-SATLIT reads 2-D grid data and creates     ***
 ***     3-D grid file; if set to 2 grid generation is by-passed    ***
 ***     and existing 3-D grid file is used.  Also, if IG(1) is     ***
 ***     set to 3 (GROUP 9) boundary conditions are calculated in   ***
 ***     SATLIT.  This current method is not fully automated,       ***
 ***     but it requires the user to examine each computational     ***
 ***     plane, which can reduce grid errors.                       ***
 ***                                                                ***
 ***********************************************************************
 *
 *
 ***********************************************************************
 ***                                                                ***
 ***   DESCRIPTION OF PLANES:  In its present form the SATLIT       ***
 ***     will write out 5 types of X-Y planes. Out of these         ***
 ***     various types of planes, modifications (or subtypes)       ***
 ***     are created (ie. the augmenter tube diameter changes).     ***
 ***     For the case that is delivered, 13 planes of grids         ***
 ***     are created.  A description of each plane is now           ***
 ***     provided.                                                  ***
 ***                                                                ***
 ***   TYPE 1  --  Indicated by letter A.  This type is used        ***
***     for the orifice.  There are five planes created under     ***
***     this type.  The first is located at the entrance.  In     ***
***     this plane the circles of the orifice and the reduction   ***
***     gear/engine have been mapped into a square.  The second   ***
***     is a cross section at the front of the orifice.  The      ***
***     third is a plane at the constant cross section of the     ***
***     orifice.  The fourth is at the start of the reduction     ***
***     gear while the fifth is located at the start of the       ***
***     engine. The last four planes contain an inner circle      ***
***     which corresponds to the diameter of reduction gear/      ***
***     engine and a mid circle which represents the prop.        ***
***     ASSUMPTION:  It is assumed that the diameter of the       ***
***     prop is less than both orifice openings.  Also, the       ***
***     outer diameter has been increase at the fourth plane.     ***
***                                                               ***
***   TYPE 2  --  Indicated by letter B.  This type is used at    ***
***     the engine exit and the lip region.  This type produces   ***
***     two or three planes of data.  Normally it will produce    ***
***     one for the augmenter lip and one for the augmenter       ***
***     sleeve.  If the engine falls in the tappered lip          ***
***     section an additional plane corresponding to this         ***
***     location will be needed and will become the second of     ***
***     the three planes produced.                                ***
***                                                               ***
***   TYPE 3  --  Indicated by letter C.  This type is used to    ***
***     create the augmenter tube in the building.  There are     ***
***     three planes produced under this type.  The first is      ***
***     located at the end of the augmenter sleeve, the second    ***
***     is a cross section of the large diameter portion of the   ***
***     tube, while the third is a slice of the small diameter    ***
***     section of the augmenter tube.                            ***
***                                                               ***
***   TYPE 4  --  Indicated by letter D.  This type is used       ***
***     to create the augmenter tube in the chimney section.      ***
***     Two planes are created for this type.  The first is for   ***
***     the start of the triangler section while the last is      ***
***     located at the end of triangler section.  ASSUMPTION:     ***
***     the point section is sliced off at given distance in      ***
***     order to make the walls fall outside the diameter of      ***
***     the augmenter tube.                                       ***
***                                                               ***
***   TYPE 5  --  Data needed for this plane is taken from the    ***
***     other types.  This type creates the exit plane (top of    ***
***     chimney).                                                 ***
***                                                               ***
***   At this time each variable used in the description of       ***
***   griding in the X-direction (horizontal) will be provided.   ***
***   The (0,0,0) coordinate is located (standing in front of     ***
***   building) at the lower right hand corner.  Parameters       ***
***   are used extensively throughout this program to make        ***
***   changes easier.                                             ***
***                                                               ***
*********************************************************************
*
*
*********************************************************************
*********                X-DIRECTION GRIDING             ************
*********************************************************************
***                                                               ***
***   NRXA    --  Number of X regions for Type 1 plane            ***
***    NRXB     --   Number of X regions for Type 2 plane             ***
***    NRXC     --   Number of X regions for Type 3 plane             ***
***    NRXD     --   Number of X regions for Type 4 plane             ***
***                                                                   ***
***    NOTE:  The number of grid cells is define for the Type 1       ***
***           plane and then redistributed for the other types.       ***
***           There are fourteen available regions, some may not      ***
***           be used.                                                ***
***                                                                   ***
***    NX01     --   Number of cells in 1st region -> Wall to         ***
***                  gap                                              ***
***    NX02     --   Number of cells in 2nd region -> Gap to          ***
***                  orifice                                          ***
***    NX03     --   Number of cells in 3rd region -> Orifice to      ***
***                  prop tip                                         ***
***    NX04     --   Number of cells in 4th region -> Prop tip to     ***
***                  center of prop                                   ***
***    NX05     --   Number of cells in 7th region -> Center of       ***
***                  prop to prop tip                                 ***
***    NX06     --   Number of cells in 8th region -> Prop tip to     ***
***                  orifice                                          ***
***    NX07     --   Number of cells in 9th region -> Orifice to      ***
***                  wall                                             ***
***    NX08     --   Number of cells in 10th region -> Spare          ***
***    NX09     --   Number of cells in 10th region -> Spare          ***
***    NX10     --   Number of cells in 10th region -> Spare          ***
***    NX11     --   Number of cells in 11th region -> Spare          ***
***    NX12     --   Number of cells in 12th region -> Spare          ***
***    NX13     --   Number of cells in 13th region -> Spare          ***
***    NX14     --   Number of cells in 14th region -> Spare          ***
***                                                                   ***
***    NOTE:  The regions for the other 4 Types will now also         ***
***           be defined.                                             ***
***    TYPE 2                                                         ***
***      Region 1  --  Wall to half distance augmenter tube           ***
***      Region 2  --  Half distance augmenter tube to aug tube       ***
***      Region 3  --  Augmenter tube to engine                       ***
***      Region 4  --  Engine to midpoint engine                      ***
***      Region 5  --  Midpoint engine to engine                      ***
***      Region 6  --  Engine to augmenter tube                       ***
***      Region 7  --  Augmenter tube to half distance aug tube       ***
***      Region 8  --  Half distance augmenter tube to wall           ***
***    TYPE 3                                                         ***
***      Region 1  --  Wall to half distance augmenter tube           ***
***      Region 2  --  Half distance augmenter tube to aug tube       ***
***      Region 3  --  Augmenter tube to midpoint aug tube            ***
***      Region 4  --  Midpoint augmenter tube to aug tube            ***
***      Region 5  --  Augmenter tube to half distance aug tube       ***
***      Region 6  --  Half distance augmenter tube to wall           ***
***    TYPE 4                                                         ***
***      Region 1  --  Wall to augmenter tube                         ***
***      Region 2  --  Augmenter tube to midpoint aug tube            ***
***      Region 3  --  Midpoint augmenter tube to aug tube            ***
***      Region 4  --  Augmenter tube to wall                         ***
***    TYPE 5                                                         ***
***      Region 1  --  Wall to wall                                   ***
***                                                                   ***
***    NXAD     --   One-half number of cells in X-direction used     ***
***                  for the reduction gear/engine                    ***
***    NXBD     --   Number of cells in X-direction used for          ***
***                 rearrangement of three regions into two        ***
***    NOTE:  This last two items have corresponding parameters    ***
***           for the Y-direction.  Generally they will be the     ***
***           the same                                             ***
***                                                                ***
***    IXAF**  --  First cell number of ** region Type 1           ***
***    IXAL**  --  Last cell number of ** region Type 1            ***
***    IXBF**  --  First cell number of ** region Type 2           ***
***    IXBL**  --  Last cell number of ** region Type 2            ***
***    IXCF**  --  First cell number of ** region Type 3           ***
***    IXCL**  --  Last cell number of ** region Type 3            ***
***    IXDF**  --  First cell number of ** region Type 4           ***
***    IXDL**  --  Last cell number of ** region Type 4            ***
***                                                                ***
***    IXMON*  --  Location of * monitoring point (9 extra)        ***
***                                                                ***
***    XLA**   --  Length to end of ** region Type 1 (in)          ***
***    XLB**   --  Length to end of ** region Type 2 (in)          ***
***    XLC**   --  Length to end of ** region Type 3 (in)          ***
***    XLD**   --  Length to end of ** region Type 4 (in)          ***
***                                                                ***
***    PXA**   --  Clustering factor of ** region Type 1           ***
***    PXB**   --  Clustering factor of ** region Type 2           ***
***    PXC**   --  Clustering factor of ** region Type 3           ***
***    PXD**   --  Clustering factor of ** region Type 4           ***
***    NOTE:  Clustering factor is a number used to shift the      ***
***           cell spacing in one direction.  This direction is    ***
***           controlled by setting this value to either a         ***
***           positive or negative value.  The default (uniform    ***
***           spacing) is 1.0.  This value may be less than or     ***
***           greater than 1.0.                                    ***
***                                                                ***
***    NOTE:  Some Y-info defined here                             ***
***    XCENA   --  Location in the X-direction of the center of    ***
***                the orifice (in)                                ***
***    YCENA   --  Location in the Y-direction of the center of    ***
***                the orifice (in)                                ***
***    XCENB   --  Location in the X-direction of the center of    ***
***                the prop and reduction gear (in)                ***
***    YCENB   --  Location in the Y-direction of the center of    ***
***                the prop and reduction gear (in)                ***
***    XCENC   --  Location in the X-direction of the center of    ***
***                the engine (in)                                 ***
***    YCENC   --  Location in the Y-direction of the center of    ***
***                the engine (in)                                 ***
***    XCEND   --  Location in the X-direction of the center of    ***
***                the augmenter tube (in)                         ***
***    YCEND   --  Location in the Y-direction of the center of    ***
***                the augmenter tube (in)                         ***
***                                                                ***
***    DORFF   --  Diameter of orifice front (in)                  ***
***    DORFB   --  Diameter of orifice back (in)                   ***
***    DPROP   --  Diameter of prop (in)                           ***
***    DGEAR   --  Diameter of reduction gear (in)                 ***
***    DENGI   --  Diameter of engine (in)                         ***
***    DAUGL   --  Diameter of augmenter tube lip (in)             ***
***    DAUGS   --  Diameter of augmenter sleeve (in)               ***
***    DAGTF   --  Diameter of aug tube before reduction (in)      ***
***    DAGTB   --  Diameter of aug tube after reduction (in)       ***
***                                                                ***
***   NOTE:  The following input is for the engine exit falling  ***
***          in the augmenter lip region.  The number of planes  ***
***          produced is controlled by the setting of IG(60).    ***
***          For this situation it will be set to 3 other wise   ***
***          it will be 2.                                       ***
***          IG, RG, & LG are built in arrays that allow for     ***
***          easy transfer of integers, reals, and logicals to   ***
***          the various modules of the code.                    ***
***   DAUGM   --  Diameter of augmenter tube lip midpoint (in)   ***
***                                                              ***
***   XGAP    --  Length in X-direction of the upper gap (in)    ***
***                                                              ***
***   IGAP    --  Number of cells in upper gap                   ***
***                                                              ***
***   PI      --  PI                                             ***
***                                                              ***
******************************************************************
*
*
*****************************************************************
***                                                           ***
***   LOGICALS:  There is 1 logical flag in the Q1 file.  It  ***
***              is outlined below.                           ***
***                                                           ***
***   LG(1)  --  T if the engine exit falls in aug lip region ***
***                                                           ***
***   WARNING:  Certain lines of coding have to be activated  ***
***             or deactivated for certain logicals.  Search  ***
***             for the string &&&LG&&& to locate such coding. ***
***             Active coding starts in the first two columns. ***
***   NOTE:  There is certain coding that is needed for       ***
***          specific grid types.  It will be ignored if not  ***
***          needed.  Generally this type of data is indented ***
***          by one space.                                    ***
***                                                           ***
*****************************************************************
*
*
*****************************************************************
***                                                           ***
***   OTHER STUFF:  Additional information is needed in the   ***
***     SATLIT to create the grid input files for the grid    ***
***     generation package.  For each type of plane in both   ***
***     the X & Y directions the user must specify what region ***
***     the 'circle' starts on.  For instance in the          ***
***     X-direction for the Type 1 it is the third region,    ***
***     therefore it is passed into SATLIT in the 17 slot     ***
***     (ie IG(117)) of the last cell number.  It is assumed  ***
***     that the first X-Coordinate is 0.0.  This is the case ***
***     in all planes except the two created for the chimney. ***
***     For these cases the first X-distance is passed to     ***
***     SATLIT through the RG array elements that are 10 above ***
***     the logical unit used to write out the grid data file. ***
***     For this case it is the 11th (LU=71) and 12th (LU=72) ***
***     planes and RG(81) and RG(82) are set to the           ***
***     appropriate values.                                   ***
***                                                           ***
*****************************************************************
*
*
```

```
  *XXXXXXXXXXXXXXXXXXXXX   DECLARE X   XXXXXXXXXXXXXXXXXXXXXXXXXXXXX*
  *
  *
INTEGER(NRXA,NRXB,NRXC,NRXD)
INTEGER(NX01,NX02,NX03,NX04,NX05,NX06,NX07,NX08,NX09,NX10)
INTEGER(NX11,NX12,NX13,NX14)
INTEGER(NXAD,NXBD)
INTEGER(IXAF01,IXAF02,IXAF03,IXAF04,IXAF05)
INTEGER(IXAF06,IXAF07,IXAF08,IXAF09,IXAF10)
INTEGER(IXAF11,IXAF12,IXAF13,IXAF14,IXAF15)
INTEGER(IXAL01,IXAL02,IXAL03,IXAL04,IXAL05)
INTEGER(IXAL06,IXAL07,IXAL08,IXAL09,IXAL10)
INTEGER(IXAL11,IXAL12,IXAL13,IXAL14,IXAL15)
INTEGER(IXBF01,IXBF02,IXBF03,IXBF04,IXBF05)
INTEGER(IXBF06,IXBF07,IXBF08,IXBF09,IXBF10)
INTEGER(IXBF11,IXBF12,IXBF13,IXBF14,IXBF15)
INTEGER(IXBL01,IXBL02,IXBL03,IXBL04,IXBL05)
INTEGER(IXBL06,IXBL07,IXBL08,IXBL09,IXBL10)
INTEGER(IXBL11,IXBL12,IXBL13,IXBL14,IXBL15)
INTEGER(IXCF01,IXCF02,IXCF03,IXCF04,IXCF05)
INTEGER(IXCF06,IXCF07,IXCF08,IXCF09,IXCF10)
INTEGER(IXCF11,IXCF12,IXCF13,IXCF14,IXCF15)
INTEGER(IXCL01,IXCL02,IXCL03,IXCL04,IXCL05)
INTEGER(IXCL06,IXCL07,IXCL08,IXCL09,IXCL10)
INTEGER(IXCL11,IXCL12,IXCL13,IXCL14,IXCL15)
INTEGER(IXDF01,IXDF02,IXDF03,IXDF04,IXDF05)
INTEGER(IXDF06,IXDF07,IXDF08,IXDF09,IXDF10)
INTEGER(IXDF11,IXDF12,IXDF13,IXDF14,IXDF15)
INTEGER(IXDL01,IXDL02,IXDL03,IXDL04,IXDL05)
INTEGER(IXDL06,IXDL07,IXDL08,IXDL09,IXDL10)
INTEGER(IXDL11,IXDL12,IXDL13,IXDL14,IXDL15)
INTEGER(IXMON1,IXMON2,IXMON3,IXMON4,IXMON5)
INTEGER(IXMON6,IXMON7,IXMON8,IXMON9)
INTEGER(ITMP1,ITMP2,IGAP)
REAL(XLA01,XLA02,XLA03,XLA04,XLA05)
REAL(XLA06,XLA07,XLA08,XLA09,XLA10)
REAL(XLA11,XLA12,XLA13,XLA14,XLA15)
REAL(XLB01,XLB02,XLB03,XLB04,XLB05)
REAL(XLB06,XLB07,XLB08,XLB09,XLB10)
REAL(XLB11,XLB12,XLB13,XLB14,XLB15)
REAL(XLC01,XLC02,XLC03,XLC04,XLC05)
REAL(XLC06,XLC07,XLC08,XLC09,XLC10)
REAL(XLC11,XLC12,XLC13,XLC14,XLC15)
REAL(XLD01,XLD02,XLD03,XLD04,XLD05)
REAL(XLD06,XLD07,XLD08,XLD09,XLD10)
REAL(XLD11,XLD12,XLD13,XLD14,XLD15)
REAL(PXA01,PXA02,PXA03,PXA04,PXA05)
REAL(PXA06,PXA07,PXA08,PXA09,PXA10)
REAL(PXA11,PXA12,PXA13,PXA14,PXA15)
REAL(PXB01,PXB02,PXB03,PXB04,PXB05)
REAL(PXB06,PXB07,PXB08,PXB09,PXB10)
REAL(PXB11,PXB12,PXB13,PXB14,PXB15)
REAL(PXC01,PXC02,PXC03,PXC04,PXC05)
REAL(PXC06,PXC07,PXC08,PXC09,PXC10)
REAL(PXC11,PXC12,PXC13,PXC14,PXC15)
REAL(PXD01,PXD02,PXD03,PXD04,PXD05)
REAL(PXD06,PXD07,PXD08,PXD09,PXD10)
REAL(PXD11,PXD12,PXD13,PXD14,PXD15)
REAL(XCENA,YCENA,XCENB,YCENB,XCENC,YCENC,XCEND,YCEND,YROCD)
REAL(DORFF,DORFB,DPROP,DGEAR,DENGI)
REAL(DAUGL,DAUGM,DAUGS,DAGTF,DAGTB)
REAL(PI,XGAP)
  *
  *
  *XXXXXXXXXXXXXXXXXXXX    LOGICALS    XXXXXXXXXXXXXXXXXXXXXXXXXXX*
  *
  *
LG(1)=T
  *
  *
  *XXXXXXXXXXXXXXXXXXXX    CIRCLE CENTERS & DIAMETERS   XXXXXXXXXXX*
  *
  *
PI=3.141592654
XCENA=140.0;               RG(41)=XCENA
YCENA=113.0;               RG(42)=YCENA
XCENB=XCENA;               RG(43)=XCENB
YCENB=YCENA;               RG(44)=YCENB
XCENC=XCENA+0.0;           RG(45)=XCENC
YCENC=YCENA+9.0;           RG(46)=YCENC
XCEND=XCENA;               RG(47)=XCEND
YCEND=YCENA;               RG(48)=YCEND
 ROCD=51.0

DORFF=189.4;               RG(50)=DORFF
DORFB=167.8;               RG(51)=DORFB
DPROP=156.0;               RG(52)=DPROP
DGEAR= 27.0;               RG(53)=DGEAR
DENGI= 18.0;               RG(54)=DENGI
DAUGL= 67.0;               RG(55)=DAUGL

   &&&LG&&&  ACTIVATE WHEN ENGINE DOES NOT FALL IN LIP  &&&1T&&&
 IG(60)=2
 DAUGS= 55.0;               RG(56)=DAUGS
   &&&LG&&&  ACTIVATE WHEN ENGINE DOES NOT FALL IN LIP  &&&1F&&&
    IG(60)=3
    DAUGM= 60.0;               RG(56)=DAUGM
    DAUGS= 55.0;               RG(57)=DAUGS

DAGTF= 56.0;               RG(58)=DAGTF
DAGTB= 34.0;               RG(59)=DAGTB
XGAP=28.0;                 RG(61)=XGAP
   #######  IGAP MUST BE EVEN  #######
IGAP=4;                    IG(61)=IGAP
  *
  *
  *XXXXXXXXXXXXXXXXXXXX    TYPE 1 DATA   XXXXXXXXXXXXXXXXXXXXXXXXXXX*
  *
  *
NRXA=7;                    IG(42)=NRXA
NXAD=4;                    IG(50)=NXAD
NX01=2
NX02=3
NX03=3
NX04=9
NX05=9
NX06=3
NX07=5

IXAF01=       1;           IXAL01=       NX01
```

```
IXAF02=IXAL01+1;          IXAL02=IXAL01+NX02
IXAF03=IXAL02+1;          IXAL03=IXAL02+NX03
IXAF04=IXAL03+1;          IXAL04=IXAL03+NX04
IXAF05=IXAL04+1;          IXAL05=IXAL04+NX05
IXAF06=IXAL05+1;          IXAL06=IXAL05+NX06
IXAF07=IXAL06+1;          IXAL07=IXAL06+NX07

XLA01= 16.000000;         PXA01= 1.0
XLA02=  0.000000;         PXA02=-1.2
XLA03=  0.000000;         PXA03= 1.0
XLA04= XCENA;             PXA04= 1.0
XLA05=  0.000000;         PXA05= 1.0
XLA06=  0.000000;         PXA06= 1.0
XLA07=280.000000;         PXA07= 1.3

IG(101)=IXAL01;RG(101)=XLA01;RG(121)=PXA01
IG(102)=IXAL02;RG(102)=XLA02;RG(122)=PXA02
IG(103)=IXAL03;RG(103)=XLA03;RG(123)=PXA03
IG(104)=IXAL04;RG(104)=XLA04;RG(124)=PXA04
IG(105)=IXAL05;RG(105)=XLA05;RG(125)=PXA05
IG(106)=IXAL06;RG(106)=XLA06;RG(126)=PXA06
IG(107)=IXAL07;RG(107)=XLA07;RG(127)=PXA07
IG(117)=3
  *
  *
  *XXXXXXXXXXXXXXXXXXXXX    TYPE 2 DATA   XXXXXXXXXXXXXXXXXXXXXXXXXXX*
  *
  *
NRXB=8;                  IG(44)=NRXB
IXBF01=IXAF01;           IXBL01=IXAL03/2
IXBF02=IXAL03/2+1;       IXBL02=IXAL03
IXBF03=IXAF04;           IXBL03=IXAL04-NXAD
IXBF04=IXAF05-NXAD;      IXBL04=IXAL04
IXBF05=IXAF05;           IXBL05=IXAL04+NXAD
IXBF06=IXAF05+NXAD;      IXBL06=IXAL05
NXBD=(IXAL07-IXAL05)/2
IXBF07=IXAF06;           IXBL07=IXAL05+NXBD
IXBF08=IXAF06+NXBD;      IXBL08=IXAL07

XLB01=  0.000000;        PXB01= 1.4
XLB02=  0.000000;        PXB02=-1.4
XLB03=  0.000000;        PXB03= 1.0
XLB04= XCENB;            PXB04= 1.0
XLB05=  0.000000;        PXB05= 1.0
XLB06=  0.000000;        PXB06= 1.0
XLB07=  0.000000;        PXB07= 1.4
XLB08= XLA07;            PXB08=-1.4

IG(141)=IXBL01;RG(181)=XLB01;RG(201)=PXB01
IG(142)=IXBL02;RG(182)=XLB02;RG(202)=PXB02
IG(143)=IXBL03;RG(183)=XLB03;RG(203)=PXB03
IG(144)=IXBL04;RG(184)=XLB04;RG(204)=PXB04
IG(145)=IXBL05;RG(185)=XLB05;RG(205)=PXB05
IG(146)=IXBL06;RG(186)=XLB06;RG(206)=PXB06
IG(147)=IXBL07;RG(187)=XLB07;RG(207)=PXB07
IG(148)=IXBL08;RG(188)=XLB08;RG(208)=PXB08
IG(157)=3
  *
  *
  *XXXXXXXXXXXXXXXXXXXXX    TYPE 3 DATA   XXXXXXXXXXXXXXXXXXXXXXXXXXX*
  *
  *
 RXC=6;                  IG(46)=NRXC
_XCF01=IXBF01;           IXCL01=IXBL01
IXCF02=IXBF02;           IXCL02=IXBL02
¯XCF03=IXBF03;           IXCL03=IXBL04
 XCF04=IXBF05;           IXCL04=IXBL06
IXCF05=IXBF07;           IXCL05=IXBL07
TXCF06=IXBF08;           IXCL06=IXBL08

..LC01=  0.000000;       PXC01= PXB01
XLC02=  0.000000;        PXC02= PXB02
 LC03= XCEND;            PXC03= 1.2
 LC04=  0.000000;        PXC04=-1.2
XLC05=  0.000000;        PXC05= PXB07
"LC06= XLA07;            PXC06= PXB08

ıG(181)=IXCL01;RG(261)=XLC01;RG(281)=PXC01
IG(182)=IXCL02;RG(262)=XLC02;RG(282)=PXC02
 G(183)=IXCL03;RG(263)=XLC03;RG(283)=PXC03
_G(184)=IXCL04;RG(264)=XLC04;RG(284)=PXC04
IG(185)=IXCL05;RG(265)=XLC05;RG(285)=PXC05
 G(186)=IXCL06;RG(266)=XLC06;RG(286)=PXC06
 G('197)=3
  *
  *
  *XXXXXXXXXXXXXXXXXXXX    TYPE 4 DATA   XXXXXXXXXXXXXXXXXXXXXXXXXXX*
  *
  *
 RXD=4;                  IG(48)=NRXD
 XDF01=IXCF01;           IXDL01=IXCL02
IXDF02=IXCF03;           IXDL02=IXCL03
¯XDF03=IXCF04;           IXDL03=IXCL04
 XDF04=IXCF05;           IXDL04=IXCL06

TTMP1=79+IG(60)
 TMP2=80+IG(60)
..G(ITMP1)=116.0
RG(ITMP2)= 92.0
 LD01=  0.000000;        PXD01=-1.2
 LD02=  XCEND;           PXD02= PXC03
XLD03=  0.000000;        PXD03= PXC04
"LD04=  0.000000;        PXD04= 1.2

ıG(221)=IXDL01;RG(341)=XLD01;RG(361)=PXD01
IG(222)=IXDL02;RG(342)=XLD02;RG(362)=PXD02
 G(223)=IXDL03;RG(343)=XLD03;RG(363)=PXD03
_G(224)=IXDL04;RG(344)=XLD04;RG(364)=PXD04
IG(237)=2
  *
  *
  ****************************************************************
  *********             Y-DIRECTION GRIDING             ************
  ****************************************************************
  ***                                                          ***
  ***   NRYA    --  Number of Y regions for Type 1 plane       ***
  ***   NRYB    --  Number of Y regions for Type 2 plane       ***
  ***   NRYC    --  Number of Y regions for Type 3 plane       ***
  ***   NRYD    --  Number of Y regions for Type 4 plane       ***
  ***                                                          ***
***    NOTE:  The number of grid cells is define for the Type 1   ***
***           plane and then redistributed for the other types.   ***
***           There are fourteen available regions, some may      ***
***           not be used.                                        ***
***                                                               ***
***    NY01     --  Number of cells in 1st region -> Floor to     ***
***                 orifice                                       ***
***    NY02     --  Number of cells in 2nd region -> Orifice to   ***
***                 prop tip                                      ***
***    NX03     --  Number of cells in 3rd region -> Prop tip to  ***
***                 center of prop                                ***
***    NX04     --  Number of cells in 6th region -> Reduction    ***
***                 gear to prop tip                              ***
***    NY05     --  Number of cells in 7th region -> Prop tip     ***
***                 to orifice                                    ***
***    NY06     --  Number of cells in 8th region -> Orifice to   ***
***                 bottom of gap                                 ***
***    NY07     --  Number of cells in 9th region -> Bottom of    ***
***                 gap to top of gap                             ***
***    NY08     --  Number of cells in 10th region -> Top of gap  ***
***                 to roof                                       ***
***    NY09     --  Number of cells in 11th region -> Spare       ***
***    NY10     --  Number of cells in 11th region -> Spare       ***
***    NY11     --  Number of cells in 11th region -> Spare       ***
***    NY12     --  Number of cells in 12th region -> Spare       ***
***    NY13     --  Number of cells in 13th region -> Spare       ***
***    NY14     --  Number of cells in 14th region -> Spare       ***
***                                                               ***
***    NOTE:  The regions for the other 4 Types will now also     ***
***           be defined.                                         ***
***    TYPE 2                                                     ***
***      Region 1  --  Floor to half distance augmenter tube      ***
***      Region 2  --  Half distance augmenter tube to aug tube   ***
***      Region 3  --  Augmenter tube to engine                   ***
***      Region 4  --  Engine to midpoint engine                  ***
***      Region 5  --  Midpoint engine to engine                  ***
***      Region 6  --  Engine to augmenter tube                   ***
***      Region 7  --  Augmenter tube to half distance aug tube   ***
***      Region 8  --  Half distance augmenter tube to roof       ***
***    TYPE 3                                                     ***
***      Region 1  --  Floor to half distance augmenter tube      ***
***      Region 2  --  Half distance augmenter tube to aug tube   ***
***      Region 3  --  Augmenter tube to midpoint aug tube        ***
***      Region 4  --  Midpoint augmenter tube to aug tube        ***
***      Region 5  --  Augmenter tube to half distance aug tube   ***
***      Region 6  --  Half distance augmenter tube to roof       ***
***    TYPE 4                                                     ***
***      Region 1  --  Floor to augmenter tube                    ***
***      Region 2  --  Augmenter tube to midpoint aug tube        ***
***      Region 3  --  Midpoint augmenter tube to aug tube        ***
***      Region 4  --  Augmenter tube to center of curvature      ***
***      Region 5  --  Center of curvature to roof                ***
***    TYPE 5                                                     ***
***      Region 1  --  Wall to wall                               ***
***                                                               ***
***    NYAD     --  One-half number of cells in Y-direction used  ***
***                 for the reduction gear/engine                 ***
***    NYBD     --  Number of cells in Y-direction used for       ***
***                 rearrangement of three regions into two       ***
***    NOTE:  This last two items have corresponding parameters   ***
```

```
  ***          for the X-direction.  Generally they will be the    ***
  ***          the same.                                           ***
  ***                                                              ***
  ***   IYAF**  --  First cell number of ** region Type 1          ***
  ***   IYAL**  --  Last cell number of ** region Type 1           ***
  ***   IYBF**  --  First cell number of ** region Type 2          ***
  ***   IYBL**  --  Last cell number of ** region Type 2           ***
  ***   IYCF**  --  First cell number of ** region Type 3          ***
  ***   IYCL**  --  Last cell number of ** region Type 3           ***
  ***   IYDF**  --  First cell number of ** region Type 4          ***
  ***   IYDL**  --  Last cell number of ** region Type 4           ***
  ***                                                              ***
  ***   IYMON*  --  Location of * monitoring point (9 extra)       ***
  ***                                                              ***
  ***   YLA**   --  Length to end of ** region Type 1 (in)         ***
  ***   YLB**   --  Length to end of ** region Type 2 (in)         ***
  ***   YLC**   --  Length to end of ** region Type 3 (in)         ***
  ***   YLD**   --  Length to end of ** region Type 4 (in)         ***
  ***                                                              ***
  ***   PYA**   --  Clustering factor of ** region Type 1          ***
  ***   PYB**   --  Clustering factor of ** region Type 2          ***
  ***   PYC**   --  Clustering factor of ** region Type 3          ***
  ***   PYD**   --  Clustering factor of ** region Type 4          ***
  ***                                                              ***
  ******************************************************************
  *
  *
  *YYYYYYYYYYYYYYYYYYYYY   DECLARE Y   YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY*
  *
  *
INTEGER(NRYA,NRYB,NRYC,NRYD)
INTEGER(NYAD,NYBD)
INTEGER(NY01,NY02,NY03,NY04,NY05,NY06,NY07,NY08,NY09,NY10)
INTEGER(NY11,NY12,NY13,NY14)
INTEGER(IYAF01,IYAF02,IYAF03,IYAF04,IYAF05)
INTEGER(IYAF06,IYAF07,IYAF08,IYAF09,IYAF10)
INTEGER(IYAF11,IYAF12,IYAF13,IYAF14,IYAF15)
INTEGER(IYAL01,IYAL02,IYAL03,IYAL04,IYAL05)
INTEGER(IYAL06,IYAL07,IYAL08,IYAL09,IYAL10)
INTEGER(IYAL11,IYAL12,IYAL13,IYAL14,IYAL15)
INTEGER(IYBF01,IYBF02,IYBF03,IYBF04,IYBF05)
INTEGER(IYBF06,IYBF07,IYBF08,IYBF09,IYBF10)
INTEGER(IYBF11,IYBF12,IYBF13,IYBF14,IYBF15)
INTEGER(IYBL01,IYBL02,IYBL03,IYBL04,IYBL05)
INTEGER(IYBL06,IYBL07,IYBL08,IYBL09,IYBL10)
INTEGER(IYBL11,IYBL12,IYBL13,IYBL14,IYBL15)
INTEGER(IYCF01,IYCF02,IYCF03,IYCF04,IYCF05)
INTEGER(IYCF06,IYCF07,IYCF08,IYCF09,IYCF10)
INTEGER(IYCF11,IYCF12,IYCF13,IYCF14,IYCF15)
INTEGER(IYCL01,IYCL02,IYCL03,IYCL04,IYCL05)
INTEGER(IYCL06,IYCL07,IYCL08,IYCL09,IYCL10)
INTEGER(IYCL11,IYCL12,IYCL13,IYCL14,IYCL15)
INTEGER(IYDF01,IYDF02,IYDF03,IYDF04,IYDF05)
INTEGER(IYDF06,IYDF07,IYDF08,IYDF09,IYDF10)
INTEGER(IYDF11,IYDF12,IYDF13,IYDF14,IYDF15)
INTEGER(IYDL01,IYDL02,IYDL03,IYDL04,IYDL05)
INTEGER(IYDL06,IYDL07,IYDL08,IYDL09,IYDL10)
INTEGER(IYDL11,IYDL12,IYDL13,IYDL14,IYDL15)
INTEGER(IYMON1,IYMON2,IYMON3,IYMON4,IYMON5)
INTEGER(IYMON6,IYMON7,IYMON8,IYMON9)
```

```
REAL(YLA01,YLA02,YLA03,YLA04,YLA05)
REAL(YLA06,YLA07,YLA08,YLA09,YLA10)
REAL(YLA11,YLA12,YLA13,YLA14,YLA15)
REAL(YLB01,YLB02,YLB03,YLB04,YLB05)
REAL(YLB06,YLB07,YLB08,YLB09,YLB10)
REAL(YLB11,YLB12,YLB13,YLB14,YLB15)
REAL(YLC01,YLC02,YLC03,YLC04,YLC05)
REAL(YLC06,YLC07,YLC08,YLC09,YLC10)
REAL(YLC11,YLC12,YLC13,YLC14,YLC15)
REAL(YLD01,YLD02,YLD03,YLD04,YLD05)
REAL(YLD06,YLD07,YLD08,YLD09,YLD10)
REAL(YLD11,YLD12,YLD13,YLD14,YLD15)
REAL(PYA01,PYA02,PYA03,PYA04,PYA05)
REAL(PYA06,PYA07,PYA08,PYA09,PYA10)
REAL(PYA11,PYA12,PYA13,PYA14,PYA15)
REAL(PYB01,PYB02,PYB03,PYB04,PYB05)
REAL(PYB06,PYB07,PYB08,PYB09,PYB10)
REAL(PYB11,PYB12,PYB13,PYB14,PYB15)
REAL(PYC01,PYC02,PYC03,PYC04,PYC05)
REAL(PYC06,PYC07,PYC08,PYC09,PYC10)
REAL(PYC11,PYC12,PYC13,PYC14,PYC15)
REAL(PYD01,PYD02,PYD03,PYD04,PYD05)
REAL(PYD06,PYD07,PYD08,PYD09,PYD10)
REAL(PYD11,PYD12,PYD13,PYD14,PYD15)
  *
  *
  *YYYYYYYYYYYYYYYYYYYY    TYPE 1 DATA    YYYYYYYYYYYYYYYYYYYYYYYYYYY*
  *
  *
NRYA=8;                    IG(43)=NRYA
NYAD=4;                    IG(51)=NYAD
NY01=5
NY02=3
NY03=9
NY04=9
NY05=3
NY06=3
NY07=2
NY08=2

IYAF01=       1;          IYAL01=       NY01
IYAF02=IYAL01+1;          IYAL02=IYAL01+NY02
IYAF03=IYAL02+1;          IYAL03=IYAL02+NY03
IYAF04=IYAL03+1;          IYAL04=IYAL03+NY04
IYAF05=IYAL04+1;          IYAL05=IYAL04+NY05
IYAF06=IYAL05+1;          IYAL06=IYAL05+NY06
IYAF07=IYAL06+1;          IYAL07=IYAL06+NY07
IYAF08=IYAL07+1;          IYAL08=IYAL07+NY08

YLA01=  0.000000;         PYA01= 1.0
YLA02=  0.000000;         PYA02= 1.0
YLA03= YCENA;             PYA03= 1.0
YLA04=  0.000000;         PYA04= 1.0
YLA05=  0.000000;         PYA05= 1.0
YLA06=238.000000;         PYA06= 1.0
YLA07=244.000000;         PYA07= 1.0
YLA08=294.000000;         PYA08= 1.0

IG(121)=IYAL01;RG(141)=YLA01;RG(161)=PYA01
IG(122)=IYAL02;RG(142)=YLA02;RG(162)=PYA02
IG(123)=IYAL03;RG(143)=YLA03;RG(163)=PYA03
IG(124)=IYAL04;RG(144)=YLA04;RG(164)=PYA04
IG(125)=IYAL05;RG(145)=YLA05;RG(165)=PYA05
IG(126)=IYAL06;RG(146)=YLA06;RG(166)=PYA06
IG(127)=IYAL07;RG(147)=YLA07;RG(167)=PYA07
IG(128)=IYAL08;RG(148)=YLA08;RG(168)=PYA08
IG(137)=2
  *
  *
  *YYYYYYYYYYYYYYYYYYYYY    TYPE 2 DATA   YYYYYYYYYYYYYYYYYYYYYYYYYYY*
  *
  *
NRYB=8;                    IG(45)=NRYB
IYBF01=IYAF01;             IYBL01=IYAL02/2
IYBF02=IYAL02/2+1;         IYBL02=IYAL02
IYBF03=IYAF03;             IYBL03=IYAL03-NYAD
IYBF04=IYAF04-NYAD;        IYBL04=IYAL03
IYBF05=IYAF04;             IYBL05=IYAL03+NYAD
IYBF06=IYAF04+NYAD;        IYBL06=IYAL04
NYBD=(IYAL08-IYAL04)/2
IYBF07=IYAF05;             IYBL07=IYAL04+NYBD
IYBF08=IYAF05+NYBD;        IYBL08=IYAL08

YLB01=  0.000000;          PYB01= 1.4
YLB02=  0.000000;          PYB02=-1.4
YLB03=  0.000000;          PYB03= 1.0
YLB04=  YCENB;             PYB04= 1.0
YLB05=  0.000000;          PYB05= 1.0
YLB06=  0.000000;          PYB06= 1.0
YLB07=  0.000000;          PYB07= 1.4
YLB08= YLA08;              PYB08=-1.4

IG(161)=IYBL01;RG(221)=YLB01;RG(241)=PYB01
IG(162)=IYBL02;RG(222)=YLB02;RG(242)=PYB02
IG(163)=IYBL03;RG(223)=YLB03;RG(243)=PYB03
IG(164)=IYBL04;RG(224)=YLB04;RG(244)=PYB04
IG(165)=IYBL05;RG(225)=YLB05;RG(245)=PYB05
IG(166)=IYBL06;RG(226)=YLB06;RG(246)=PYB06
IG(167)=IYBL07;RG(227)=YLB07;RG(247)=PYB07
IG(168)=IYBL08;RG(228)=YLB08;RG(248)=PYB08
IG(177)=3
  *
  *
  *YYYYYYYYYYYYYYYYYYYYY    TYPE 3 DATA   YYYYYYYYYYYYYYYYYYYYYYYYYYY*
  *
  *
NRYC=6;                    IG(47)=NRYC
IYCF01=IYBF01;             IYCL01=IYBL01
IYCF02=IYBF02;             IYCL02=IYBL02
 YCF03=IYBF03;             IYCL03=IYBL04
 YCF04=IYBF05;             IYCL04=IYBL06
IYCF05=IYBF07;             IYCL05=IYBL07
IYCF06=IYBF08;             IYCL06=IYBL08

YLC01=  0.000000;          PYC01= PYB01
YLC02=  0.000000;          PYC02= PYB02
YLC03=  YCEND;             PYC03= 1.2
YLC04=  0.000000;          PYC04=-1.2
YLC05=  0.000000;          PYC05= PYB07
 LC06= YLA08;              PYC06= PYB08
```

```
IG(201)=IYCL01;RG(301)=YLC01;RG(321)=PYC01
IG(202)=IYCL02;RG(302)=YLC02;RG(322)=PYC02
IG(203)=IYCL03;RG(303)=YLC03;RG(323)=PYC03
IG(204)=IYCL04;RG(304)=YLC04;RG(324)=PYC04
IG(205)=IYCL05;RG(305)=YLC05;RG(325)=PYC05
IG(206)=IYCL06;RG(306)=YLC06;RG(326)=PYC06
IG(217)=3
  *
  *
  *YYYYYYYYYYYYYYYYYYYYY    TYPE 4 DATA   YYYYYYYYYYYYYYYYYYYYYYYYYYY*
  *
  *
NRYD=4;                 IG(49)=NRYD
IYDF01=IYCF01;          IYDL01=IYCL02
IYDF02=IYCF03;          IYDL02=IYCL03
IYDF03=IYCF04;          IYDL03=IYCL04
IYDF04=IYCF05;          IYDL04=IYCL06

YLD01=  0.000000;       PYD01=-1.4
YLD02=  YCEND;          PYD02= PYC03
YLD03=  0.000000;       PYD03= PYC04
YLD04=  YCEND+YROCD;    PYD04= 1.2

IG(241)=IYDL01;RG(381)=YLD01;RG(401)=PYD01
IG(242)=IYDL02;RG(382)=YLD02;RG(402)=PYD02
IG(243)=IYDL03;RG(383)=YLD03;RG(403)=PYD03
IG(244)=IYDL04;RG(384)=YLD04;RG(404)=PYD04
IG(257)=2
  *
  *
  ********************************************************************
  *********              Z-DIRECTION GRIDING              ************
  ********************************************************************
  ***                                                              ***
  ***   NOTE:  With the X-Y gird information, several planes of    ***
  ***          grid points will be produced in the SATLIT.  In     ***
  ***          this section the user must specify how these        ***
  ***          planes are then stacked, blended, or rotated.       ***
  ***          There will be a plane of data for the front face    ***
  ***          of each of the following regions.                   ***
  ***                                                              ***
  ***   NCS     --  Number of regions in Z-direction               ***
  ***                                                              ***
  ***                                                              ***
  ***   NZ01    --  Number of cells in 1st region -> End of        ***
  ***               baffles to half distance orifice               ***
  ***   NZ02    --  Number of cells in 2nd region -> Half          ***
  ***               distance orifice to start of orifice           ***
  ***   NZ03    --  Number of cells in 3rd region -> Start of      ***
  ***               orifice to orifice angle                       ***
  ***   NZ04    --  Number of cells in 4th region -> Orifice       ***
  ***               angle to end of orifice                        ***
  ***   NZ05    --  Number of cells in 5th region -> End of        ***
  ***               orifice to start of prop                       ***
  ***   NZ06    --  Number of cells in 6th region -> Start of      ***
  ***               prop to end of prop                            ***
  ***   NZ07    --  Number of cells in 7th region -> End of prop   ***
  ***               to start of reduction gear                     ***
  ***   NZ08    --  Number of cells in 8th region -> Start of      ***
***                    reduction gear to end of reduction gear          ***
***     NZ09     --    Number of cells in 9th region -> End of          ***
***                    reduction gear to start of engine                ***
***     NZ10     --    Number of cells in 10th region -> Start of       ***
***                    engine to end of engine and augmenter lip        ***
***     NZ11     --    Number of cells in 11th region -> Augmenter      ***
***                    lip to start of augmenter sleeve                 ***
***     NZ12     --    Number of cells in 12th region -> Start of       ***
***                    augmenter sleeve to end augmenter sleeve         ***
***     NZ13     --    Number of cells in 13th region -> End of         ***
***                    augmenter sleeve to augmenter tube               ***
***                    ASSUMPTION:  This is an arbitrary region to      ***
***                    make up for the difference in diameter.          ***
***     NZ14     --    Number of cells in 20th region -> Augmenter      ***
***                    tube to start of augmenter tube reduction        ***
***     NZ15     --    Number of cells in 15th region -> Start of       ***
***                    augmenter tube reduction to end of reduction     ***
***     NZ16     --    Number of cells in 16th region -> End of         ***
***                    augmenter tube reduction to start of baffles     ***
***     NZ17     --    Number of cells in 17th region -> Start of       ***
***                    baffles to start of triangular room              ***
***                    ASSUMPTION:  The start of the triangular room    ***
***                    has be 'chopped' off for orthogonality           ***
***     NZ18     --    Number of cells in 18th region -> Start of       ***
***                    triangular room to start of chimney              ***
***     NZ19     --    Number of cells in 19th region -> Start of       ***
***                    of chimney to end of augmenter tube              ***
***     NZ20     --    Number of cells in 20th region -> End of         ***
***                    augmenter tube to end of domain                  ***
***     NZ21     --    Number of cells in 21th region -> Spare          ***
***     NZ22     --    Number of cells in 22th region -> Spare          ***
***     NZ23     --    Number of cells in 23th region -> Spare          ***
***     NZ24     --    Number of cells in 24th region -> Spare          ***
***     NZ25     --    Number of cells in 25th region -> Spare          ***
***                                                                     ***
***     IZF**    --    First cell number of ** region                   ***
***     IZL**    --    Last cell number of ** region                    ***
***                                                                     ***
***     IZMON*   --    Location of * monitoring point (9 extra)         ***
***                                                                     ***
***     ZL**     --    Length to end of ** region (in)                  ***
***                                                                     ***
***     PZ**     --    Clustering factor of ** region                   ***
***                                                                     ***
***     ITRI     --    Number of cells in straight section of           ***
***                    augmenter tube in chimney                        ***
***                                                                     ***
***     ZPT      --    Length of straight section of augmenter tube     ***
***                    in chimney                                       ***
***                                                                     ***
***     ZCH      --    Height at start of baffles in chimney            ***
***                                                                     ***
***************************************************************************
*
*
*ZZZZZZZZZZZZZZZZZZZZ   DECLARE Z   ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ*
*
*
INTEGER(NCS,ITRI)
 NTEGER(NZ01,NZ02,NZ03,NZ04,NZ05,NZ06,NZ07,NZ08,NZ09,NZ10)
```

```
INTEGER(NZ11,NZ12,NZ13,NZ14,NZ15,NZ16,NZ17,NZ18,NZ19,NZ20)
INTEGER(NZ21,NZ22,NZ23,NZ24,NZ25)
INTEGER(IZF01,IZF02,IZF03,IZF04,IZF05)
INTEGER(IZF06,IZF07,IZF08,IZF09,IZF10)
INTEGER(IZF11,IZF12,IZF13,IZF14,IZF15)
INTEGER(IZF16,IZF17,IZF18,IZF19,IZF20)
INTEGER(IZF21,IZF22,IZF23,IZF24,IZF25)
INTEGER(IZL01,IZL02,IZL03,IZL04,IZL05)
INTEGER(IZL06,IZL07,IZL08,IZL09,IZL10)
INTEGER(IZL11,IZL12,IZL13,IZL14,IZL15)
INTEGER(IZL16,IZL17,IZL18,IZL19,IZL20)
INTEGER(IZL21,IZL22,IZL23,IZL24,IZL25)
INTEGER(IZMON1,IZMON2,IZMON3,IZMON4,IZMON5)
INTEGER(IZMON6,IZMON7,IZMON8,IZMON9)
REAL(ZL01,ZL02,ZL03,ZL04,ZL05)
REAL(ZL06,ZL07,ZL08,ZL09,ZL10)
REAL(ZL11,ZL12,ZL13,ZL14,ZL15)
REAL(ZL16,ZL17,ZL18,ZL19,ZL20)
REAL(ZL21,ZL22,ZL23,ZL24,ZL25)
REAL(PZ01,PZ02,PZ03,PZ04,PZ05)
REAL(PZ06,PZ07,PZ08,PZ09,PZ10)
REAL(PZ11,PZ12,PZ13,PZ14,PZ15)
REAL(PZ16,PZ17,PZ18,PZ19,PZ20)
REAL(PZ21,PZ22,PZ23,PZ24,PZ25)
REAL(ZPT,ZCH)
  *
  *
  *ZZZZZZZZZZZZZZZZZZZZZ    GEOMETRY & STACKING INFO  ZZZZZZZZZZZZZZZ*
  *
  *
NCS=20;                      IG(501)=NCS
NZ01=4
NZ02=4
NZ03=4
NZ04=2
NZ05=3
NZ06=1
NZ07=4
NZ08=3
NZ09=3
NZ10=5
NZ11=2
NZ12=5
NZ13=1
NZ14=4
NZ15=3
NZ16=4
NZ17=5
NZ18=2
   #######  DIFFERENCE BETWEEN NZ19 AND ITRI MUST BE EVEN #######
ITRI=1;                      IG(90)=ITRI
NZ19=11
NZ20=5

ZPT=9.0;                     RG(90)=ZPT
ZCH=317.0;                   RG(91)=ZCH

IZF01=       1;              IZL01=       NZ01
IZF02=IZL01+1;               IZL02=IZL01+NZ02
IZF03=IZL02+1;               IZL03=IZL02+NZ03
```

```
IZF04=IZL03+1;          IZL04=IZL03+NZ04
IZF05=IZL04+1;          IZL05=IZL04+NZ05
IZF06=IZL05+1;          IZL06=IZL05+NZ06
IZF07=IZL06+1;          IZL07=IZL06+NZ07
IZF08=IZL07+1;          IZL08=IZL07+NZ08
IZF09=IZL08+1;          IZL09=IZL08+NZ09
IZF10=IZL09+1;          IZL10=IZL09+NZ10
IZF11=IZL10+1;          IZL11=IZL10+NZ11
IZF12=IZL11+1;          IZL12=IZL11+NZ12
IZF13=IZL12+1;          IZL13=IZL12+NZ13
IZF14=IZL13+1;          IZL14=IZL13+NZ14
IZF15=IZL14+1;          IZL15=IZL14+NZ15
IZF16=IZL15+1;          IZL16=IZL15+NZ16
IZF17=IZL16+1;          IZL17=IZL16+NZ17
IZF18=IZL17+1;          IZL18=IZL17+NZ18
IZF19=IZL18+1;          IZL19=IZL18+NZ19
IZF20=IZL19+1;          IZL20=IZL19+NZ20

ZL01=102.5;             PZ01= 1.3
ZL02=205.0;             PZ02=-1.6
ZL03=215.8;             PZ03= 1.0
ZL04=217.8;             PZ04= 1.0
   note prop width 5" ass
ZL05=231.3;             PZ05= 1.0
ZL06=236.3;             PZ06= 1.0
ZL07=247.8;             PZ07= 1.2
ZL08=267.8;             PZ08= 1.0
ZL09=298.8;             PZ09= 1.0
ZL10=394.8;             PZ10= 1.0
ZL11=400.8;             PZ11= 1.0
ZL12=430.8;             PZ12= 1.0
ZL13=436.8;             PZ13= 1.0
ZL14=532.8;             PZ14= 1.0
ZL15=603.8;             PZ15= 1.0
ZL16=689.8;             PZ16= 1.0
ZL17=842.3;             PZ17= 1.0
ZL18=871.8;             PZ18= 1.0
ZL19=991.8;             PZ19= 1.0
ZL20=991.8;             PZ20= 1.4

IG(511)=NZ01;RG(511)=ZL01;RG(541)=PZ01
IG(512)=NZ02;RG(512)=ZL02;RG(542)=PZ02
IG(513)=NZ03;RG(513)=ZL03;RG(543)=PZ03
IG(514)=NZ04;RG(514)=ZL04;RG(544)=PZ04
IG(515)=NZ05;RG(515)=ZL05;RG(545)=PZ05
IG(516)=NZ06;RG(516)=ZL06;RG(546)=PZ06
IG(517)=NZ07;RG(517)=ZL07;RG(547)=PZ07
IG(518)=NZ08;RG(518)=ZL08;RG(548)=PZ08
IG(519)=NZ09;RG(519)=ZL09;RG(549)=PZ09
IG(520)=NZ10;RG(520)=ZL10;RG(550)=PZ10
IG(521)=NZ11;RG(521)=ZL11;RG(551)=PZ11
IG(522)=NZ12;RG(522)=ZL12;RG(552)=PZ12
IG(523)=NZ13;RG(523)=ZL13;RG(553)=PZ13
IG(524)=NZ14;RG(524)=ZL14;RG(554)=PZ14
IG(525)=NZ15;RG(525)=ZL15;RG(555)=PZ15
IG(526)=NZ16;RG(526)=ZL16;RG(556)=PZ16
IG(527)=NZ17;RG(527)=ZL17;RG(557)=PZ17
IG(528)=NZ18;RG(528)=ZL18;RG(558)=PZ18
IG(529)=NZ19;RG(529)=ZL19;RG(559)=PZ19
IG(530)=NZ20;RG(530)=ZL20;RG(560)=PZ20
```

```
   #######  IG WHERE CHIMNEY STARTS  #######
IG(537)=19

IG(541)=1;IG(571)=61;IG(601)=61
IG(542)=2;IG(572)=61;IG(602)=62
IG(543)=2;IG(573)=62;IG(603)=63
IG(544)=1;IG(574)=63;IG(604)=63
IG(545)=1;IG(575)=63;IG(605)=63
IG(546)=1;IG(576)=63;IG(606)=63
IG(547)=2;IG(577)=63;IG(607)=64
IG(548)=1;IG(578)=64;IG(608)=64
IG(549)=2;IG(579)=64;IG(609)=65
IG(550)=2;IG(580)=65;IG(610)=66
IG(551)=2;IG(581)=66;IG(611)=67
IG(552)=2;IG(582)=67;IG(612)=68
IG(553)=2;IG(583)=68;IG(613)=69
IG(554)=1;IG(584)=69;IG(614)=69
IG(555)=2;IG(585)=69;IG(615)=70
IG(556)=1;IG(586)=70;IG(616)=70
IG(557)=2;IG(587)=70;IG(617)=71
IG(558)=2;IG(588)=71;IG(618)=72
IG(559)=3;IG(589)=72;IG(619)=72
IG(560)=4;IG(590)=72;IG(620)=73

    GROUP 2. Transience; time-step specification
    GROUP 3. X-direction grid specification
NX=NX01+NX02+NX03+NX04+NX05+NX06+NX07+NX08+NX09
NX=NX+NX10+NX11+NX12+NX13+NX14
    GROUP 4. Y-direction grid specification
NY=NY01+NY02+NY03+NY04+NY05+NY06+NY07+NY08+NY09
NY=NY+NY10+NY11+NY12+NY13+NY14
    GROUP 5. Z-direction grid specification
NZ=NZ01+NZ02+NZ03+NZ04+NZ05+NZ06+NZ07+NZ08+NZ09+NZ10+NZ11
NZ=NZ+NZ12+NZ13+NZ14+NZ15+NZ16+NZ17+NZ18+NZ19+NZ20+NZ21
NZ=NZ+NZ22+NZ23+NZ24+NZ25
    GROUP 6. Body-fitted coordinates or grid distortion
BFC=T;NONORT=T
IG(1)=2
SATRUN(NECL)
READCO(GRID)
    GROUP 7. Variables stored, solved & named
SOLUTN(P1,Y,Y,Y,N,N,N)
SOLVE(U1,V1,W1)
SOLUTN(U1,Y,Y,N,Y,N,N)
SOLUTN(V1,Y,Y,N,Y,N,N)
SOLUTN(W1,Y,Y,N,Y,N,N)
SOLVE(H1,C1)
STORE(RHO1)
STORE(C3,C4,C5,C6,C7)
STORE(U2,V2,W2,C8,C9,C10,C11)
NAME(C4)=TEMP
NAME(C5)=CP
NAME(C8)=PH2O
NAME(C9)=TFAR
NAME(C10)=RHOE
NAME(C11)=SPAR
TURMOD(KEMODL)
STORE(ENUT)
KELIN=1
```

```
    GROUP 8. Terms (in differential equations) & devices
TERMS(H1,N,P,P,P,P,P)
    GROUP 9. Properties of the medium (or media)
  *
  *********************************************************************
  *********            USER DEFINED VARIABLES                ************
  *********************************************************************
  ***                                                               ***
  ***   NOTE:  These are the variables used to define this          ***
  ***          problem.                                             ***
  ***                                                               ***
  ***   COND1   --   k for mineral fiber (BTU/hr/ft/F)              ***
  ***   COND2   --   k for steel (BTU/hr/ft/F)                      ***
  ***   THICK1  --   Thickness of mineral fiber (in)                ***
  ***   THICK2  --   Thickness of steel (in)                        ***
  ***   TAMB    --   Temperature ambient (F)                        ***
  ***   TENG    --   Temperature engine (F)                         ***
  ***   EMDOT   --   Engine flow rate (lb/s)                        ***
  ***   FMDOT   --   Fuel flow rate (lb/s)                          ***
  ***   PAMB    --   Pressure ambient (mm Hg)                       ***
  ***   TIG     --   Turbulence intensity inlet (-)                 ***
  ***   TIE     --   Turbulence intensity engine (-)                ***
  ***   XKFCT1  --   K-loss factor inlet (-)                        ***
  ***   XKFCT2  --   K-loss factor outlet (-)                       ***
  ***   XKFCT3  --   K-loss factor chimney (-)                      ***
  ***   AMF1    --   N2 mass fraction ambient (-)                   ***
  ***   AMF2    --   O2 mass fraction ambient (-)                   ***
  ***   AMF3    --   CO2 mass fraction ambient (-)                  ***
  ***   AMF4    --   H2O mass fraction ambient (-)                  ***
  ***   EMF1    --   N2 mass fraction engine (-)                    ***
  ***   EMF2    --   O2 mass fraction engine (-)                    ***
  ***   EMF3    --   CO2 mass fraction engine (-)                   ***
  ***   EMF4    --   H2O mass fraction engine (-)                   ***
  ***   RPM     --   RPM of turboprop (r/m)                         ***
  ***   SHP     --   Shaft horse power of engine (hp)               ***
  ***   PCTK    --   % of engine power wasted generating            ***
  ***                turbulence (%)                                 ***
  ***   ICURVE  --   Selector for Ct/Cp curve in ground (1 or 2)    ***
  ***   NRAMP   --   # sweeps over which to ramp in KE-EP prop      ***
  ***                sources at beginning of calculation (1-20)     ***
  ***                                                               ***
  ***   WARNING:  There are temperature traps set in GROUP 18. At   ***
  ***             the present time the values are 273 K and 950 K. ***
  ***                                                               ***
  *********************************************************************
  *
  *
  *********************************************************************
  *********              OTHER VARIABLES                     ************
  *********************************************************************
  ***                                                               ***
  ***   NOTE:  These are the variables used to define this          ***
  ***          problem.                                             ***
  ***                                                               ***
  ***   XCON01  --   Converts in to m                               ***
  ***   XCON02  --   Converts F to R                                ***
  ***   XCON03  --   Converts R to K                                ***
  ***   XCON04  --   Converts BTU/ft/h/R to J/s/m/K                 ***
  ***   XCON05  --   Converts lb to kg                              ***
  ***   XCON06  --   Converts N/sq m to in H2O                      ***
  ***   XCON07  --  Converts m/s to ft/s                           ***
  ***   XCON08  --  Converts kg/cu m to lb/cu ft                   ***
  ***   XCON09  --  Converts in Hg to N/sq m                       ***
  ***   XCON10  --  Spare                                          ***
  ***   XCON11  --  Spare                                          ***
  ***   PTRAP   --  Pressure trap (N/sq m)                         ***
  ***   RGAS    --  Gas constant  (N-m/K/kgmol)                    ***
  ***   XMW1    --  Molecular weight N2 (kg/kgmol)                 ***
  ***   XMW2    --  Molecular weight O2 (kg/kgmol)                 ***
  ***   XMW3    --  Molecular weight CO2 (kg/kgmol)                ***
  ***   XMW4    --  Molecular weight H2O (kg/kgmol)                ***
  ***   EARSOR  --  Area of engine plate source side (sq m)        ***
  ***   RHOAMB  --  Density ambient (kg/cu m)                      ***
  ***   RHOENG  --  Density engine (kg/cu m)                       ***
  ***   ENTHA   --  Enthalpy ambient (J/kg)                        ***
  ***   ENTHE   --  Enthalpy engine (J/kg)                         ***
  ***   GAPIN   --  Gap between inlet baffles (m)                  ***
  ***   GKE     --  Inlet KE (sq m/sq s)                           ***
  ***   GEP     --  Inlet EP (sq m/cu s)                           ***
  ***   EKE     --  Engine KE (sq m/sq s)                          ***
  ***   EEP     --  Engine EP (sq m/cu s)                          ***
  ********************************************************************
  *
REAL(TAMB,RGAS,TENG)
REAL(EARSOR,RHOAMB,RHOENG)
REAL(AMF1,AMF2,AMF3,AMF4,EMF1,EMF2,EMF3,EMF4)
REAL(ENTHA,ENTHE,XMW1,XMW2,XMW3,XMW4,XMWA,XMWE)
REAL(COND1,THICK1,COND2,THICK2,PTRAP)
REAL(XCON01,XCON02,XCON03,XCON04,XCON05,XCON06)
REAL(XCON07,XCON08,XCON09,XCON10,XCON11)
REAL(EMDOT,FMDOT,PAMB)
REAL(GAPIN,TIG,TIE,GKE,GEP,EKE,EEP)
REAL(XKFCT1,XKFCT2,XKFCT3)
REAL(RPM,SHP,PCTK)
INTEGER(ICURVE,NRAMP)
  *
  ********************************************************************
  *********                   USER SECTION                   *********
  ********************************************************************
  *
COND1=0.022
COND2=26.0
THICK1=2.0
THICK2=0.25
TAMB=77.0
TENG=1100.0
EMDOT=32.4
FMDOT=0.8333333
PAMB=29.92
TIG=0.02
TIE=0.15
XKFCT1=1.0
XKFCT2=1.0
XKFCT3=1.0
AMF1=0.7683
AMF2=0.2317
AMF3=0.0
AMF4=0.0
EMF1=0.7479
EMF2=0.1411
```

```
EMF3=0.081
EMF4=0.030
 PM=1021.
 HP=4591.
PCTK=5.
 CURVE=2
 RAMP=10
  *
  ***********************************************************************
  *
      conversions
XCON01=0.0254;           RG(31)=XCON01
 CON02=459.67;           RG(32)=XCON02
 CON03=5.0/9.0;          RG(33)=XCON03
XCON04=1.73073;          RG(34)=XCON04
XCON05=0.45359;          RG(35)=XCON05
 CON06=407.16/101325.0;  RG(36)=XCON06
XCON07=3.2802;           RG(37)=XCON07
XCON08=0.062428;         RG(38)=XCON08
 CON09=101325.0/29.92;   RG(39)=XCON09

COND1=COND1*XCON04
COND2=COND2*XCON04
 HICK1=THICK1*XCON01
THICK2=THICK2*XCON01
TAMB=(TAMB+XCON02)*XCON03
 ENG=(TENG+XCON02)*XCON03
EMDOT=EMDOT*XCON05
FMDOT=FMDOT*XCON05
 G(702)=FMDOT
 AMB=PAMB*XCON09
      density info
 TRAP=0.05
 G(29)=PTRAP
RGAS=8314.32
RG( 1)=AMF1
 G( 2)=AMF2
RG( 3)=AMF3
RG( 4)=AMF4
 G( 5)=EMF1
 G( 6)=EMF2
RG( 7)=EMF3
RG( 8)=EMF4
 G( 9)=TAMB
RG(10)=TENG
XMW1=28.1608;            RG(21)=XMW1
 MW2=31.9988;            RG(22)=XMW2
XMW3=44.0100;            RG(23)=XMW3
XMW4=18.0152;            RG(24)=XMW4
 G(25)=RGAS
      area calculation engine
EARSOR=PI*(DENGI/2.*XCON01)*(DENGI/2.*XCON01)
RG(802)=EARSOR
      density calculation
XMWA=1.0/(AMF1/XMW1+AMF2/XMW2+AMF3/XMW3+AMF4/XMW4)
XMWE=1.0/(EMF1/XMW1+EMF2/XMW2+EMF3/XMW3+EMF4/XMW4)
 HOAMB=PAMB*XMWA/(RGAS*TAMB)
RG(701)=RHOAMB
RHOENG=PAMB*XMWE/(RGAS*TENG)
      run satlit for enthalpy calculation
```

```
IG(1)=3
SATRUN(NECL)
     other stuff
ENTHA=RG(11)
ENTHE=RG(12)
PRESS0=PAMB
RHO1=GRND
DRH1DP=GRND
     turbulence  (assume 1 ft gap and 1 m/s velocity)
GAPIN=1.0*XCON01
GKE=0.5*(1.0*TIG)**2
GEP=0.164*GKE**1.5/(0.09*GAPIN)
EKE=0.5*((EMDOT/EARSOR/RHOENG)*TIE)**2
EEP=0.164*EKE**1.5/(0.09*EARSOR**0.5)

   GROUP 10. Inter-phase-transfer processes and properties
  *
  ********************************************************************
  *********                   INDEX                          *********
  ********************************************************************
  ***                                                              ***
  ***   The following variables are used as an index to define     ***
  ***   the extent of blockages in the X, Y, & Z directions.       ***
  ***   This was done because a user may change the number of      ***
  ***   regions in each direction.  The user will then make the    ***
  ***   appropriate changes in this section and then no further    ***
  ***   changes will be required below this section.  The          ***
  ***   nomenclature for the variables below is as follows:        ***
  ***   1.)  The first letter represents direction (ie I for X),   ***
  ***   2.)  Middle two letters represents the blockage name, &    ***
  ***   3.)  Last letter represents first or last.                 ***
  ***                                                              ***
  ********************************************************************
  *
INTEGER(IO1F,IO1L,JO1F,JO1L,KO1F,KO1L)
INTEGER(IO2F,IO2L,JO2F,JO2L,KO2F,KO2L)
INTEGER(IO3F,IO3L,JO3F,JO3L,KO3F,KO3L)
INTEGER(IO4F,IO4L,JO4F,JO4L,KO4F,KO4L)
INTEGER(IO5F,IO5L,JO5F,JO5L,KO5F,KO5L)
INTEGER(IO6F,IO6L,JO6F,JO6L,KO6F,KO6L)
INTEGER(IO7F,IO7L,JO7F,JO7L,KO7F,KO7L)
INTEGER(IPRF,IPRL,JPRF,JPRL,KPRF,KPRL)
INTEGER(IRGF,IRGL,JRGF,JRGL,KRGF,KRGL)
INTEGER(IEGF,IEGL,JEGF,JEGL,KEGF,KEGL)
INTEGER(IA1F,IA1L,JA1F,JA1L,KA1F,KA1L)
INTEGER(IA2F,IA2L,JA2F,JA2L,KA2F,KA2L)
INTEGER(IW1F,IW1L,JW1F,JW1L,KW1F,KW1L)
INTEGER(IW2F,IW2L,JW2F,JW2L,KW2F,KW2L)
INTEGER(IW3F,IW3L,JW3F,JW3L,KW3F,KW3L)
INTEGER(IW4F,IW4L,JW4F,JW4L,KW4F,KW4L)
INTEGER(IIN,KEP)
INTEGER(IEPF,IEPL,JEPF,JEPL)

   orifice housing (lower section)
IO1F=IXAF01;             IO1L=IXAL07
JO1F=IYAF01;             JO1L=IYAL01
KO1F=IZF03;              KO1L=IZL04
   orifice housing (mid-right section)
IO2F=IXAF01;             IO2L=IXAL02
JO2F=IYAF02;             JO2L=IYAL05
```

```
KO2F=IZF03;                 KO2L=IZL04
    orifice housing (mid-left section)
IO3F=IXAF07;                IO3L=IXAL07
JO3F=IYAF02;                JO3L=IYAL05
KO3F=IZF03;                 KO3L=IZL04
    orifice housing (above orifice-below gap)
IO4F=IXAF01;                IO4L=IXAL07
JO4F=IYAF06;                JO4L=IYAL06
KO4F=IZF03;                 KO4L=IZL04
    orifice housing (beside gap)
IO5F=IXAF01;                IO5L=IXAL01
JO5F=IYAF07;                JO5L=IYAL07
KO5F=IZF03;                 KO5L=IZL04
    orifice housing (top-right section)
IO6F=IXAF01;                IO6L=IXAL04-IGAP/2
JO6F=IYAF08;                JO6L=IYAL08
KO6F=IZF03;                 KO6L=IZL04
    orifice housing (top-left section)
IO7F=IXAF05+IGAP/2;         IO7L=IXAL07
JO7F=IYAF08;                JO7L=IYAL08
KO7F=IZF03;                 KO7L=IZL04

    prop
IPRF=IXAF04;                IPRL=IXAL05
JPRF=IYAF03;                JPRL=IYAL04
KPRF=IZF06;                 KPRL=IZL06

    reduction gear
IRGF=IXAF05-NXAD;           IRGL=IXAL04+NXAD
JRGF=IYAF04-NYAD;           JRGL=IYAL03+NYAD
KRGF=IZF08;                 KRGL=IZL08

    engine
IEGF=IXAF05-NXAD;           IEGL=IXAL04+NXAD
JEGF=IYAF04-NYAD;           JEGL=IYAL03+NYAD
KEGF=IZF10;                 KEGL=IZL10
    engine plate
IEPF=IEGF;                  IEPL=IEGL
JEPF=JEGF;                  JEPL=JEGL
KEP=IZF10+2

IG(711)=KEP
IG(712)=IEPF;IG(713)=IEPL
IG(714)=JEPF;IG(715)=JEPL

    augmenter tube (in building)
IA1F=IXCF03;                IA1L=IXCL04
JA1F=IYCF03;                JA1L=IYCL04
KA1F=IZF11;                 KA1L=IZL16
    augmenter tube (in chimney)
IA2F=IXDF02;                IA2L=IXDL03
JA2F=IYDF02;                JA2L=IYDL03
KA2F=IZF18;                 KA2L=IZL19

    wall (lower section)
IW1F=IXDF01;                IW1L=IXDL04
JW1F=IYDF01;                JW1L=IYDL01
KW1F=IZF17;                 KW1L=IZL18
    wall (mid-right section)
IW2F=IXDF01;                IW2L=IXDL01
```

```
JW2F=IYDF02;              JW2L=IYDL03
KW2F=IZF17;               KW2L=IZL18
    wall (mid-left section)
IW3F=IXDF04;              IW3L=IXDL04
JW3F=IYDF02;              JW3L=IYDL03
KW3F=IZF17;               KW3L=IZL18
    wall (top section)
IW4F=IXDF01;              IW4L=IXDL04
JW4F=IYDF04;              JW4L=IYDL04
KW4F=IZF17;               KW4L=IZL18

    GROUP 11. Initialization of variable or porosity fields
    orifice
CONPOR(0.0,CELL,  IO1F, IO1L, JO1F,-JO1L,-KO1F,-KO1L)
CONPOR(0.0,CELL,  IO2F,-IO2L, JO2F, JO2L,-KO2F,-KO2L)
CONPOR(0.0,CELL, -IO3F, IO3L, JO3F, JO3L,-KO3F,-KO3L)
CONPOR(0.0,CELL,  IO4F, IO4L,-JO4F,-JO4L,-KO4F,-KO4L)
CONPOR(0.0,CELL,  IO5F,-IO5L, JO5F, JO5L,-KO5F,-KO5L)
CONPOR(0.0,CELL,  IO6F,-IO6L,-JO6F, JO6L,-KO6F,-KO6L)
CONPOR(0.0,CELL, -IO7F, IO7L,-JO7F, JO7L,-KO7F,-KO7L)

    reduction gear
CONPOR(0.0,CELL, -IRGF,-IRGL,-JRGF,-JRGL,-KRGF,-KRGL)

    engine
CONPOR(0.0,SOUTH, IEGF, IEGL,-JEGF,-JEGF, KEGF, KEGL)
CONPOR(0.0,NORTH, IEGF, IEGL,-JEGL,-JEGL, KEGF, KEGL)
CONPOR(0.0,WEST, -IEGF,-IEGF, JEGF, JEGL, KEGF, KEGL)
CONPOR(0.0,EAST, -IEGL,-IEGL, JEGF, JEGL, KEGF, KEGL)

    engine plate
CONPOR(0.0,HIGH,  IEPF, IEPL, JEPF, JEPL, KEP, KEP)

    augmenter tube (in building)
CONPOR(0.0,SOUTH, IA1F, IA1L,-JA1F,-JA1F, KA1F, KA1L)
CONPOR(0.0,NORTH, IA1F, IA1L,-JA1L,-JA1L, KA1F, KA1L)
CONPOR(0.0,WEST, -IA1F,-IA1F, JA1F, JA1L, KA1F, KA1L)
CONPOR(0.0,EAST, -IA1L,-IA1L, JA1F, JA1L, KA1F, KA1L)

    end wall
CONPOR(0.0,CELL,  IW1F, IW1L, JW1F,-JW1L,-KW1F,-KW1L)
CONPOR(0.0,CELL,  IW2F,-IW2L, JW2F, JW2L,-KW2F,-KW2L)
CONPOR(0.0,CELL, -IW3F, IW3L, JW3F, JW3L,-KW3F,-KW3L)
CONPOR(0.0,CELL,  IW4F, IW4L,-JW4F, JW4L,-KW4F,-KW4L)

    augmenter tube (in chimney)
CONPOR(0.0,SOUTH, IA2F, IA2L,-JA2F,-JA2F, KA2F, KA2L)
CONPOR(0.0,NORTH, IA2F, IA2L,-JA2L,-JA2L, KA2F, KA2L)
CONPOR(0.0,WEST, -IA2F,-IA2F, JA2F, JA2L, KA2F, KA2L)
CONPOR(0.0,EAST, -IA2L,-IA2L, JA2F, JA2L, KA2F, KA2L)

    init all
FIINIT(H1)=ENTHA
FIINIT(TEMP)=TAMB
FIINIT(RHO1)=RHOAMB
FIINIT(C3)=1.0
FIINIT(W1)=7.0
    init eng
PATCH(INITA,INIVAL,IEGF,IEGL,JEGF,JEGL,KEP+1,KA1F-1,1,1)
INIT (INITA,H1,0.0,ENTHE)
INIT (INITA,TEMP,0.0,TENG)
INIT (INITA,RHO1,0.0,RHOENG)
INIT (INITA,C1,0.0,1.0)
INIT (INITA,KE,0.0,EKE)
INIT (INITA,EP,0.0,EEP)
INIT (INITA,W1,0.0,EMDOT/EARSOR/RHOENG)
PATCH(INITB,INIVAL,IA1F,IA1L,JA1F,JA1L,KA1F,NZ,1,1)
INIT (INITB,H1,0.0,ENTHE)
INIT (INITB,TEMP,0.0,TENG)
INIT (INITB,RHO1,0.0,RHOENG)
INIT (INITB,C1,0.0,1.0)
INIT (INITB,KE,0.0,EKE)
INIT (INITB,EP,0.0,EEP)
PATCH(INITC,INIVAL,IA1F,IA1L,JA1F,JA1L,KA1F,NZ-1,1,1)
INIT (INITC,W1,0.0,EMDOT/EARSOR/RHOENG)

    GROUP 12. Convection and diffusion adjustments
    GROUP 13. Boundary conditions and special sources
    top wall
PATCH(XWALL01,NWALL,1,NX,NY,NY,IZF01,KW1F-1,1,1)
COVAL(XWALL01,U1,GRND2,0.0)
COVAL(XWALL01,W1,GRND2,0.0)
COVAL(XWALL01,KE,GRND2,GRND2)
COVAL(XWALL01,EP,GRND2,GRND2)
   bottom wall
PATCH(XWALL02,SWALL,1,NX,1,1,1,KW1F-1,1,1)
COVAL(XWALL02,U1,GRND2,0.0)
COVAL(XWALL02,W1,GRND2,0.0)
COVAL(XWALL02,KE,GRND2,GRND2)
COVAL(XWALL02,EP,GRND2,GRND2)
   side to block wall
PATCH(XWALL03,WWALL,1,1,1,NY,1,KW1F-1,1,1)
COVAL(XWALL03,V1,GRND2,0.0)
COVAL(XWALL03,W1,GRND2,0.0)
COVAL(XWALL03,KE,GRND2,GRND2)
COVAL(XWALL03,EP,GRND2,GRND2)
PATCH(XWALL04,EWALL,NX,NX,1,NY,1,KW1F-1,1,1)
COVAL(XWALL04,V1,GRND2,0.0)
COVAL(XWALL04,W1,GRND2,0.0)
COVAL(XWALL04,KE,GRND2,GRND2)
COVAL(XWALL04,EP,GRND2,GRND2)
    chimney wall
PATCH(XWALL05,NWALL,1,NX,NY,NY,KA2L+1,NZ,1,1)
COVAL(XWALL05,U1,GRND2,0.0)
COVAL(XWALL05,W1,GRND2,0.0)
COVAL(XWALL05,KE,GRND2,GRND2)
COVAL(XWALL05,EP,GRND2,GRND2)
PATCH(XWALL06,SWALL,1,NX,1,1,KW1L+1,NZ,1,1)
COVAL(XWALL06,U1,GRND2,0.0)
COVAL(XWALL06,W1,GRND2,0.0)
COVAL(XWALL06,KE,GRND2,GRND2)
COVAL(XWALL06,EP,GRND2,GRND2)
PATCH(XWALL07,WWALL,1,1,1,NY,KW1L+1,NZ,1,1)
COVAL(XWALL07,V1,GRND2,0.0)
COVAL(XWALL07,W1,GRND2,0.0)
COVAL(XWALL07,KE,GRND2,GRND2)
COVAL(XWALL07,EP,GRND2,GRND2)
PATCH(XWALL08,EWALL,NX,NX,1,NY,KW1L+1,NZ,1,1)
COVAL(XWALL08,V1,GRND2,0.0)
COVAL(XWALL08,W1,GRND2,0.0)
COVAL(XWALL08,KE,GRND2,GRND2)
COVAL(XWALL08,EP,GRND2,GRND2)
    front opening
PATCH(XOPEN1,LOW,1,NX,1,NY,1,1,1,1)
COVAL(XOPEN1,P1,GRND7,0.0)
COVAL(XOPEN1,W1,ONLYMS,GRND7)
COVAL(XOPEN1,H1,ONLYMS,ENTHA)
COVAL(XOPEN1,KE,ONLYMS,GKE)
COVAL(XOPEN1,EP,ONLYMS,GEP)
COVAL(XOPEN1,UCRT,ONLYMS,XKFCT1)
    back opening
PATCH(XOPEN2A,HIGH,IW1F,IW1L,JW1F,JW1L,KW1F-1,KW1F-1,1,1)
COVAL(XOPEN2A,P1,GRND7,0.0)
COVAL(XOPEN2A,W1,ONLYMS,SAME)
COVAL(XOPEN2A,H1,ONLYMS,ENTHA)
COVAL(XOPEN2A,UCRT,ONLYMS,XKFCT2)
PATCH(XOPEN2B,HIGH,IW2F,IW2L,JW2F,JW2L,KW1F-1,KW1F-1,1,1)
COVAL(XOPEN2B,P1,GRND7,0.0)
COVAL(XOPEN2B,W1,ONLYMS,SAME)
COVAL(XOPEN2B,H1,ONLYMS,ENTHA)
COVAL(XOPEN2B,UCRT,ONLYMS,XKFCT2)
PATCH(XOPEN2C,HIGH,IW3F,IW3L,JW3F,JW3L,KW1F-1,KW1F-1,1,1)
COVAL(XOPEN2C,P1,GRND7,0.0)
COVAL(XOPEN2C,W1,ONLYMS,SAME)
COVAL(XOPEN2C,H1,ONLYMS,ENTHA)
COVAL(XOPEN2C,UCRT,ONLYMS,XKFCT2)
PATCH(XOPEN2D,HIGH,IW4F,IW4L,JW4F,JW4L,KW1F-1,KW1F-1,1,1)
COVAL(XOPEN2D,P1,GRND7,0.0)
COVAL(XOPEN2D,W1,ONLYMS,SAME)
COVAL(XOPEN2D,H1,ONLYMS,ENTHA)
COVAL(XOPEN2D,UCRT,ONLYMS,XKFCT2)
    chimney exhaust
PATCH(XOPEN3,HIGH,1,NX,1,NY,NZ,NZ,1,1)
COVAL(XOPEN3,P1,GRND7,0.0)
COVAL(XOPEN3,W1,ONLYMS,SAME)
COVAL(XOPEN3,H1,ONLYMS,ENTHA)
COVAL(XOPEN3,UCRT,ONLYMS,XKFCT3)
    engine mdot sink
PATCH(XENGIN,HIGH,IEGF,IEGL,JEGF,JEGL,KEP,KEP,1,1)
COVAL(XENGIN,P1,FIXFLU,GRND10)
COVAL(XENGIN,H1,ONLYMS,SAME)
RG(804)=(EMDOT-FMDOT)/EARSOR
    engine mdot source
PATCH(XENGOUT,LOW,IEGF,IEGL,JEGF,JEGL,KEP+1,KEP+1,1,1)
COVAL(XENGOUT,P1,FIXFLU,GRND10)
COVAL(XENGOUT,W1,ONLYMS,EMDOT/EARSOR/RHOENG)
COVAL(XENGOUT,H1,ONLYMS,ENTHE)
COVAL(XENGOUT,C1,ONLYMS,1.0)
COVAL(XENGOUT,KE,ONLYMS,EKE)
COVAL(XENGOUT,EP,ONLYMS,EEP)
RG(805)=EMDOT/EARSOR
    prop
PATCH(ZPROP,PHASEM,IPRF  ,IPRL,JPRF  ,JPRL,KPRF,KPRF,1,1)
COVAL(ZPROP,W1,FIXFLU,GRND9)
PATCH(XPROP, LOW  ,IPRF-1,IPRL,JPRF  ,JPRL,KPRF,KPRF,1,1)
COVAL(XPROP,U1,FIXFLU,GRND9)
PATCH(YPROP, LOW  ,IPRF  ,IPRL,JPRF-1,JPRL,KPRF,KPRF,1,1)
COVAL(YPROP,V1,FIXFLU,GRND9)
PATCH(KPROP, LOW  ,IPRF  ,IPRL,JPRF  ,JPRL,KPRF,KPRF,1,1)
COVAL(KPROP,KE,FIXFLU,GRND9)
COVAL(KPROP,EP,FIXFLU,GRND9)
RG(830)=RPM;RG(831)=SHP
IG(832)=PCTK
IG(875)=ICURVE;IG(876)=NRAMP
    heat transfer augmenter tube (in building)
PATCH(HEATTR1E,EWALL,IA1F-1,IA1F,JA1F,JA1L,KA1F,KA1L,1,1)
COVAL(HEATTR1E,H1,GRND8,GRND8);COVAL(HEATTR1E,UCRT,COND1,THICK1)
PATCH(HEATTR1W,WWALL,IA1L,IA1L+1,JA1F,JA1L,KA1F,KA1L,1,1)
COVAL(HEATTR1W,H1,GRND8,GRND8);COVAL(HEATTR1W,UCRT,COND1,THICK1)
PATCH(HEATTR1N,NWALL,IA1F,IA1L,JA1F-1,JA1F,KA1F,KA1L,1,1)
COVAL(HEATTR1N,H1,GRND8,GRND8);COVAL(HEATTR1N,UCRT,COND1,THICK1)
PATCH(HEATTR1S,SWALL,IA1F,IA1L,JA1L,JA1L+1,KA1F,KA1L,1,1)
COVAL(HEATTR1S,H1,GRND8,GRND8);COVAL(HEATTR1S,UCRT,COND1,THICK1)
    heat transfer augmenter tube (in chimney)
PATCH(HEATTR2E,EWALL,IA2F-1,IA2F,JA2F,JA2L,KA2F,KA2L,1,1)
COVAL(HEATTR2E,H1,GRND8,GRND8);COVAL(HEATTR2E,UCRT,COND2,THICK2)
PATCH(HEATTR2W,WWALL,IA2L,IA2L+1,JA2F,JA2L,KA2F,KA2L,1,1)
COVAL(HEATTR2W,H1,GRND8,GRND8);COVAL(HEATTR2W,UCRT,COND2,THICK2)
PATCH(HEATTR2N,NWALL,IA2F,IA2L,JA2F-1,JA2F,KA2F,KA2L,1,1)
COVAL(HEATTR2N,H1,GRND8,GRND8);COVAL(HEATTR2N,UCRT,COND2,THICK2)
PATCH(HEATTR2S,SWALL,IA2F,IA2L,JA2L,JA2L+1,KA2F,KA2L,1,1)
COVAL(HEATTR2S,H1,GRND8,GRND8);COVAL(HEATTR2S,UCRT,COND2,THICK2)

    GROUP 14. Downstream pressure for PARAB=.TRUE.
FSWEEP=1
LSWEEP=2500
  *
  ****************************************************************
  *********                USER CONTROLS                 *********
  ****************************************************************
  ***                                                          ***
  ***   The following integer arrays are described below.      ***
  ***                                                          ***
  ***                                                          ***
  ***   IG(901)  --  Frequency of ground printout on wall heat ***
  ***                transfer & convergence.                   ***
  ***   IG(902)  --  Frequency of restart files and English unit ***
  ***                calculation (NOTE:  Overwrites previous). ***
  ***   IG(999)  --  Set to 1 to stop run on first sweep.      ***
  ***   IG( 38)  --  Set to 1 for first set of spot value info.  ***
  ***   IG( 39)  --  Set to 1 for second set of spot value info. ***
  ***   IG( 40)  --  Set to 1 for third set of spot value info.  ***
  ***   IG( 41)  --  Set to 1 for additional heat transfer info. ***
  ***                                                          ***
  ****************************************************************
IG(901)=50
IG(902)=100
IG(999)=0
    GROUP 15. Termination of sweeps
    GROUP 16. Termination of iterations
LITER(P1)=30
ENDIT(P1)=1.0E-3
ENDIT(H1)=1.0E-2
RESREF(P1)=1.0E-8
RESREF(U1)=1.0E-8
RESREF(V1)=1.0E-8
RESREF(W1)=1.0E-8
RESREF(H1)=1.0E-8
RESREF(C1)=1.0E-8
RESREF(KE)=1.0E-8
RESREF(EP)=1.0E-8
    GROUP 17. Under-relaxation devices
RELAX(P1,LINRLX,0.10)
RELAX(KE,LINRLX,0.10)
RELAX(EP,LINRLX,0.10)
RELAX(U1,FALSDT,0.001)
RELAX(V1,FALSDT,0.001)
RELAX(W1,FALSDT,0.001)
RELAX(H1,FALSDT,0.005)
RELAX(C1,FALSDT,0.005)
    GROUP 18. Limits on variables or increments to them
VARMAX(C1)=1.00;VARMIN(C1)=1.0E-10
VARMAX(ENUT)=10000000.*ENUL
VARMAX(TEMP)=950.0;VARMIN(TEMP)=273.0
    GROUP 19. Data communicated by satellite to GROUND
    GROUP 20. Preliminary print-out
    GROUP 21. Print-out of variables
OUTPUT(P1,Y,Y,N,Y,Y,Y)
OUTPUT(U1,Y,N,N,Y,Y,Y)
OUTPUT(V1,Y,N,N,Y,Y,Y)
OUTPUT(W1,Y,N,N,Y,Y,Y)
OUTPUT(KE,N,N,N,Y,Y,Y)
OUTPUT(EP,N,N,N,Y,Y,Y)
OUTPUT(H1,N,N,N,Y,Y,Y)
OUTPUT(C1,N,N,N,Y,Y,Y)
OUTPUT(C3,N,N,N,N,N,N)
OUTPUT(TEMP,Y,N,N,N,N,N)
OUTPUT(CP,N,N,N,N,N,N)
OUTPUT(C6,N,N,N,N,N,N)
OUTPUT(C7,N,N,N,N,N,N)
OUTPUT(U2,N,N,N,N,N,N)
OUTPUT(V2,N,N,N,N,N,N)
OUTPUT(W2,N,N,N,N,N,N)
OUTPUT(PH2O,N,N,N,N,N,N)
OUTPUT(TFAR,N,N,N,N,N,N)
OUTPUT(RHOE,N,N,N,N,N,N)
OUTPUT(SPAR,N,N,N,N,N,N)
OUTPUT(RHO1,Y,N,N,N,N,N)
OUTPUT(UCRT,N,N,N,N,N,N)
OUTPUT(VCRT,N,N,N,N,N,N)
OUTPUT(WCRT,N,N,N,N,N,N)
    GROUP 22. Spot-value print-out
IXMON =18;IYMON =18;IZMON = 3
IXMON1=28;IYMON1=28;IZMON1=12
IXMON2= 6;IYMON2= 6;IZMON2=19
IXMON3=18;IYMON3=18;IZMON3=19
IXMON4=13;IYMON4=13;IZMON4=37
IXMON5=13;IYMON5=13;IZMON5=44
IXMON6=18;IYMON6=18;IZMON6=46
IXMON7=11;IYMON7=11;IZMON7=58
IXMON8=11;IYMON8=26;IZMON8=70
IXMON9=19;IYMON9=16;IZMON9=72
IG(11)=IXMON1;IG(12)=IYMON1;IG(13)=IZMON1
IG(14)=IXMON2;IG(15)=IYMON2;IG(16)=IZMON2
IG(17)=IXMON3;IG(18)=IYMON3;IG(19)=IZMON3
IG(20)=IXMON4;IG(21)=IYMON4;IG(22)=IZMON4
IG(23)=IXMON5;IG(24)=IYMON5;IG(25)=IZMON5
IG(26)=IXMON6;IG(27)=IYMON6;IG(28)=IZMON6
IG(29)=IXMON7;IG(30)=IYMON7;IG(31)=IZMON7
IG(32)=IXMON8;IG(33)=IYMON8;IG(34)=IZMON8
IG(35)=IXMON9;IG(36)=IYMON9;IG(37)=IZMON9
IG(38)=1
IG(39)=1
IG(40)=0
IG(41)=0
    GROUP 23. Field print-out and plot control
YZPR=T;IXPRF=19;IXPRL=19
TSTSWP=5;          NPRMON=5
NPRINT=LSWEEP;     IPLTL=LSWEEP;    ITABL=3
ABSIZ=.8;          ORSIZ=.8;        NUMCLS=10
NPLT=10
    GROUP 24. Dumps for restarts
  RESTRT(ALL);NAMFI=INXS
STOP
```

APPENDIX C

```
C THIS IS THE MAIN PROGRAM OF THE SATELLITE
      PROGRAM MAIN
C FILE NAME satlit.f 09/27/87
C
C   (C) COPYRIGHT 1984, LAST REVISION 1987.
C   CONCENTRATION HEAT AND MOMENTUM LTD. ALL RIGHTS RESERVED.
C   This subroutine and the remainder of the PHOENICS code are
C   proprietary software owned by Concentration Heat and Momentum
C   Limited, 40 High Street, Wimbledon, London SW19 5AU, England.
C
      LOGICAL TALK,RUN,LVAL
      EXTERNAL WAYOUT
C
C 1   Set dimensions of blank-COMMON arrays here. WARNING: the
C     corresponding blank-COMMON arrays in subroutine SATLIT must
C     have the same dimensions.
      PARAMETER (NXFD=1000,NYFD=1000,NZFD=1000,NTFD=10000)
      PARAMETER (NTCVD=25000,NBFCD=500000)
      COMMON TCVDA(NTCVD),XFRAC(NXFD),YFRAC(NYFD),ZFRAC(NZFD),
     1TFRAC(NTFD),BFCS(NBFCD)
C
C 2   Set dimensions of PATCH-name array and the instruction-stack
C     array here. The dimension of the array NLN must be the same
C     as that of STACK. WARNING: the array NAMPAT in the MAIN
C     program of EARTH (see GROUND) must have the same dimension.
C     These are specified by the parameters npatd and nld, set below.
      PARAMETER (NPATD=1000,NLD=2000)
      COMMON/NPAT/NAMPAT(NPATD)/NSTCK/STACK(NLD)/LINENO/NLN(NLD)
      CHARACTER NAMPAT*8,STACK*72
      COMMON/CNFG/CNFIG
      CHARACTER CNFIG*48
C
C 3   Set dimension of run array to MAXRUN.
      PARAMETER (NRUND=500)
      COMMON/RUNS/RUN(NRUND)
C
C 4   Set dimensions of data-for-GROUND arrays here. WARNING: the
C     corresponding arrays in the MAIN program of EARTH (see
C     GROUND) must have the same dimensions.
      PARAMETER (NLGD=1000,NIGD=1000,NRGD=10000,NCGD=1000)
      COMMON/LGRND/LG(NLGD)/IGRND/IG(NIGD)/RGRND/RG(NRGD)
      COMMON/CGRND/CG(NCGD)
      LOGICAL LG
      CHARACTER*4 CG
C
C 5   Set dimensions of data-for-GREX1 arrays here. WARNING: the
C     corresponding arrays in the MAIN program of EARTH (see
C     GROUND) must have the same dimensions.
      COMMON/LSG/LSGD(20)/ISG/ISGD(20)/RSG/RSGD(100)/CSG/CSGD(10)
      LOGICAL LSGD
      CHARACTER*4 CSGD
C
C 6   Set dimensions for user-declared PIL variables here.
      PARAMETER (NIPD=1000,NRPD=1000)
      COMMON/NIDEC/INDEC(NIPD)/IDEC/INVAL(NIPD)
      COMMON/NRDEC/REDEC(NRPD)/RDEC/REVAL(NRPD)
      CHARACTER REDEC*6,INDEC*6
C
C 7   For more than the default of 80 variables increase nvd.
C     WARNING: the corresponding parameter nvd in the MAIN program of
```

```
C     EARTH (see ground.f) must be the same.
      PARAMETER (NVD=80)
      COMMON/LDB1/DBGPHI(NVD)/IDA1/ITERMS(NVD)/IDA2/LITER(NVD)
     1/IDA3/IORCVF(NVD)/IDA4/IORCVL(NVD)/IDA5/ISLN(NVD)/IDA6/IPRN(NVD)
     1/HDA1/NAME(NVD)/RDA1/DTFALS(NVD)/RDA2/RESREF(NVD)
     1/RDA3/PRNDTL(NVD)/RDA4/PRT(NVD)/RDA5/ENDIT(NVD)/RDA6/VARMIN(NVD)
     1/RDA7/VARMAX(NVD)/RDA8/FIINIT(NVD)/RDA9/PHINT(NVD)
     1/RDA10/CINT(NVD)/RDA11/EX(NVD)
     1/IPIP1/IP1(NVD)/HPIP2/IHP2(NVD)/RPIP1/RVAL(NVD)
     1/LPIP1/LVAL(NVD)
C
C 8   Set dimension indicators to correspond with above dimensions.
      CALL SUB4(MAXTCV,NTCVD,MAXRUN,NRUND,NBFC,NBFCD,NUMPHI,NVD)
      CALL SUB4(NLG,NLGD,NIG,NIGD,NRG,NRGD,NCG,NCGD)
      CALL SUB4(NLSG,20,NISG,20,NRSG,100,NCSG,10)
      CALL SUB4(NIPIL,NIPD,NRPIL,NRPD,NPNAM,NPATD,NSTACK,NLD)
      CALL SUB4(NXFR,NXFD,NYFR,NYFD,NZFR,NZFD,NTFR,NTFD)
C
C 9   Logical unit numbers & file names.
      CALL CNFGZZ(1)
      CALL OPENFL(6)
      CALL OPENFL(5)
      CALL READQ1(TALK,RUN,MAXRUN)
C
      CALL SMAIN1(TALK,MAXTCV,MAXRUN,NBFC,NUMPHI,NLG,NIG,NRG,NCG,
     1NLSG,NISG,NRSG,NCSG,NIPIL,NRPIL,NPNAM,NSTACK,NXFR,NYFR,NZFR,
     1NTFR)
      CALL WAYOUT(0)
      END
C**************************************************************
      SUBROUTINE SAT
C
      INCLUDE 'satear'
      INCLUDE 'satloc'
C---- Call satellite used in BFC test-battery.
      CALL BFCTST
C---- the users USERST subroutine.
          IF(NAMSAT.EQ.'USER') CALL USERST
C---- Call the SATLIT subroutine.
      CALL SATLIT
      RETURN
      END
C**************************************************************
      SUBROUTINE BFCTST
C
      INCLUDE 'satear'
      INCLUDE 'satloc'
      PARAMETER (NLGD=1000,NIGD=1000,NRGD=10000,NCGD=1000)
      COMMON/LGRND/LG(NLGD)/IGRND/IG(NIGD)/RGRND/RG(NRGD)
      COMMON/CGRND/CG(NCGD)
      LOGICAL LG
C
C---- Special sequence for BFC test battery : IG(1)=28
C
         IF(.NOT.(BFC.AND.IG(1).EQ.28.AND.IGR.EQ.1)) RETURN
      L1=MIN0(IG(2),NZ)
         IF(L1.LT.1) GO TO 2
        DO 1 IZ=1,L1
    1 CALL XCYIZ(IZ,LG(10))
    2 L2=MAX0(1,IG(3))
```

```
         IF(L2.GT.NZ) RETURN
        DO 3 IZ=L2,NZ
 3      CALL XCYIZ(IZ,LG(10))
        RETURN
        END
        SUBROUTINE USERST
        CALL WRIT40('DUMMY SUBROUTINE USERST CALLED.        ')
        RETURN
        END
C************************************************************
        SUBROUTINE SATLIT
C
        INCLUDE 'satear'
        INCLUDE 'satloc'
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION STARTS:
C
C 1     Set dimensions of blank-COMMON arrays here to the
C       dimensions of the same arrays in the MAIN program of the
C       satellite.
        PARAMETER (NXFD=1000,NYFD=1000,NZFD=1000,NTFD=10000)
        PARAMETER (NTCVD=25000,NBFCD=500000)
        COMMON TCVDA(NTCVD),XFRAC(NXFD),YFRAC(NYFD),ZFRAC(NZFD),
       1TFRAC(NTFD),BFCS(NBFCD)
C
C 2     Set dimensions of data-for-GROUND arrays here. WARNING: the
C       corresponding arrays in the MAIN program of the
C       satellite program and the EARTH program must have the same
C       dimensions.
        PARAMETER (NLGD=1000,NIGD=1000,NRGD=10000,NCGD=1000)
        COMMON/LGRND/LG(NLGD)/IGRND/IG(NIGD)/RGRND/RG(NRGD)
        COMMON/CGRND/CG(NCGD)
        LOGICAL LG
        CHARACTER*4 CG
C
C 3     Introduce SATLIT-only commons, arrays, equivalences.
C
        DIMENSION SC(4),IX(16),XL(16),XP(16),IY(16),YL(16),YP(16),
       &          NZC(26),ZL(26),ZP(26),IZT(26),IZF1(26),IZF2(26),
       &          XAS(2500),YAS(2500),ZAS(2500),XAS1(2500),YAS1(2500),
       &          ZAS1(2500),XAS2(2500),YAS2(2500),ZAS2(2500),ZASL(100)
C
C 4     User places his data statements here.
C
        GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,
       122,23,24),IGR
C
C--- GROUP 1. Run title and other preliminaries
    1 CONTINUE
      WRITE(6,*)'                          IN SATLIT '
      RETURN
C
C--- GROUP 2. Transience; time-step specification
    2 CONTINUE
      RETURN
C
C--- GROUP 3. X-direction grid specification
    3 CONTINUE
      RETURN
C
C--- GROUP 4. Y-direction grid specification
```

```
    4 CONTINUE
      RETURN
C
C--- GROUP 5. Z-direction grid specification
    5 CONTINUE
C
      IF(IG(1).GE.2) RETURN
      IF(IG(1).EQ.0) WRITE(6,*)' CREATING GRID INPUT FILES'
      IF(IG(1).GE.1) WRITE(6,*)' CALCULATING INLET LOCATION'
C
C*****************************************************************
C-pd---This is the second option for exit of the engine.  There will--
C---   will be either 2 or 3 cross sections written here depending----
C---   on the location exit.  It will be 2 if it ends before the------
C---   augmenter tube or if it ends at the start of the tappered------
C---   section or at the start of straight section.  It will be 3-----
C---   if it falls in the tappered section or after the start of------
C---   the straight section.----------------------------------------
C    t1
      DO 561 I=1,5
C
      NI=14
      IX(1 )=1
      CALL SETIV(IX,IG,100,1,NI)
C
      XCENA=RG(41)
      YCENA=RG(42)
      XCENB=RG(43)
      YCENB=RG(44)
      RAD1=RG(52)/2.
      DXI=(RAD1*RAD1/2.)**0.5
      RAD2=RG(50)/2.
      IF(I.EQ.3) RAD2=RG(51)/2.
      DXII=(RAD2*RAD2/2.)**0.5
      IFST=IG(117)
      JFST=IG(137)
C
C-pd---Do trig--------------------------------------------------
C
      DXI02=DXI+(YCENB-YCENA)
      TETT02=ASIN(DXI02/RAD2)*180./3.141592654
      DXI16=DXI-(YCENB-YCENA)
      TETT16=ASIN(DXI16/RAD2)*180./3.141592654
      DXI04=DXI+(XCENB-XCENA)
      TETT04=ASIN(DXI04/RAD2)*180./3.141592654
      DXI06=DXI-(XCENB-XCENA)
      TETT06=ASIN(DXI06/RAD2)*180./3.141592654
      DXI08=DXI+(YCENB-YCENA)
      TETT08=ASIN(DXI08/RAD2)*180./3.141592654
      DXI10=DXI-(YCENB-YCENA)
      TETT10=ASIN(DXI10/RAD2)*180./3.141592654
      DXI12=DXI-(XCENB-XCENA)
      TETT12=ASIN(DXI12/RAD2)*180./3.141592654
      DXI14=DXI+(XCENB-XCENA)
      TETT14=ASIN(DXI14/RAD2)*180./3.141592654
C
      XL(1 )=0.0
      CALL SETRV(XL,RG,100,1,NI)
      XL(IFST+1)=XCENB-DXI
      XL(IFST+3)=XCENB+DXI
      XL(IFST   )=XCENA-DXII
      XL(IFST+4)=XCENA+DXII

      CALL SETRV(XP,RG,120,2,NI)
C
      IY(1 )=1
      CALL SETIV(IY,IG,120,1,NI)
C
      YL(1 )=0.0
      CALL SETRV(YL,RG,140,1,NI)
      YL(JFST+1)=YCENB-DXI
      YL(JFST+3)=YCENB+DXI
      YL(JFST   )=YCENA-DXII
      YL(JFST+4)=YCENA+DXII
C
      CALL SETRV(YP,RG,160,2,NI)

      LU=60+I
      CG(LU)='CS  '
      I10=LU/10
      I1=LU-I10*10
      WRITE(CG(LU)(3:3),'(I1)') I10
      WRITE(CG(LU)(4:4),'(I1)') I1
      OPEN(LU,FILE=CG(LU),FORM='FORMATTED',STATUS='UNKNOWN')
      IF(RG(LU+10).NE.0.0) XL(1)=RG(LU+10)
      IRX=IG(42)
      IRY=IG(43)
      CALL WRTSQ(LU,NX,NY,IRX,IRY,IX,IY,XL,YL,XP,YP)
      IF(I.EQ.1)  THEN

C-pd---Overwrite line info with box data------------------------------
C
        IFSR=IG(117)+1
        JFSR=IG(137)+1
        IMID=IFSR+1
        JMID=JFSR+1
        IAD=IG(50)
        JAD=IG(51)
        RAD3=RG(53)/2.

        WRITE(LU,*)
        WRITE(LU,102)IX(IMID)-IAD,IX(IMID),IY(JMID)-JAD,IY(JMID)-JAD,
     &                XCENB-RAD3,YCENB-RAD3,XCENB,YCENB-RAD3,1.0
        WRITE(LU,102)IX(IMID),IX(IMID)+IAD,IY(JMID)-JAD,IY(JMID)-JAD,
     &                XCENB,YCENB-RAD3,XCENB+RAD3,YCENB-RAD3,1.0
        WRITE(LU,102)IX(IMID)-IAD,IX(IMID),IY(JMID)+JAD,IY(JMID)+JAD,
     &                XCENB-RAD3,YCENB+RAD3,XCENB,YCENB+RAD3,1.0
        WRITE(LU,102)IX(IMID),IX(IMID)+IAD,IY(JMID)+JAD,IY(JMID)+JAD,
     &                XCENB,YCENB+RAD3,XCENB+RAD3,YCENB+RAD3,1.0
        WRITE(LU,102)IX(IMID)-IAD,IX(IMID)-IAD,IY(JMID)-JAD,IY(JMID),
     &                XCENB-RAD3,YCENB-RAD3,XCENB-RAD3,YCENB,1.0
        WRITE(LU,102)IX(IMID)-IAD,IX(IMID)-IAD,IY(JMID),IY(JMID)+JAD,
     &                XCENB-RAD3,YCENB,XCENB-RAD3,YCENB+RAD3,1.0
        WRITE(LU,102)IX(IMID)+IAD,IX(IMID)+IAD,IY(JMID)-JAD,IY(JMID),
     &                XCENB+RAD3,YCENB-RAD3,XCENB+RAD3,YCENB,1.0
        WRITE(LU,102)IX(IMID)+IAD,IX(IMID)+IAD,IY(JMID),IY(JMID)+JAD,
     &                XCENB+RAD3,YCENB,XCENB+RAD3,YCENB+RAD3,1.0
C
C-pd---Shuffle lines---------------------------------------------------
```

```
        WRITE(LU,*)
        WRITE(LU,102)IX(IMID-1),IX(IMID)-IAD,IY(JMID),IY(JMID),
     &               XCENB-DXI,YCENB,XCENB-RAD3,YCENB,1.0
        WRITE(LU,102)IX(IMID)-IAD,IX(IMID),IY(JMID),IY(JMID),
     &               XCENB-RAD3,YCENB,XCENB,YCENB,1.0
        WRITE(LU,102)IX(IMID),IX(IMID)+IAD,IY(JMID),IY(JMID),
     &               XCENB,YCENB,XCENB+RAD3,YCENB,1.0
        WRITE(LU,102)IX(IMID)+IAD,IX(IMID+1),IY(JMID),IY(JMID),
     &               XCENB+RAD3,YCENB,XCENB+DXI,YCENB,1.0
        WRITE(LU,102)IX(IMID),IX(IMID),IY(JMID-1),IY(JMID)-JAD,
     &               XCENB,YCENB-DXI,XCENB,YCENB-RAD3,1.0
        WRITE(LU,102)IX(IMID),IX(IMID),IY(JMID)-JAD,IY(JMID),
     &               XCENB,YCENB-RAD3,XCENB,YCENB,1.0
        WRITE(LU,102)IX(IMID),IX(IMID),IY(JMID),IY(JMID)+JAD,
     &               XCENB,YCENB,XCENB,YCENB+RAD3,1.0
        WRITE(LU,102)IX(IMID),IX(IMID),IY(JMID)+JAD,IY(JMID+1),
     &               XCENB,YCENB+RAD3,XCENB,YCENB+DXI,1.0
C
        WRITE(LU,*)
        WRITE(LU,103)IX(IMID-1),IX(IMID)-IAD,IY(JMID-1),IY(JMID)-JAD
        WRITE(LU,103)IX(IMID)-IAD,IX(IMID),IY(JMID-1),IY(JMID)-JAD
        WRITE(LU,103)IX(IMID),IX(IMID)+IAD,IY(JMID-1),IY(JMID)-JAD
        WRITE(LU,103)IX(IMID)+IAD,IX(IMID+1),IY(JMID-1),IY(JMID)-JAD
        WRITE(LU,103)IX(IMID-1),IX(IMID)-IAD,IY(JMID)-JAD,IY(JMID)
        WRITE(LU,103)IX(IMID)-IAD,IX(IMID),IY(JMID)-JAD,IY(JMID)
        WRITE(LU,103)IX(IMID),IX(IMID)+IAD,IY(JMID)-JAD,IY(JMID)
        WRITE(LU,103)IX(IMID)+IAD,IX(IMID+1),IY(JMID)-JAD,IY(JMID)
        WRITE(LU,103)IX(IMID-1),IX(IMID)-IAD,IY(JMID),IY(JMID)+JAD
        WRITE(LU,103)IX(IMID)-IAD,IX(IMID),IY(JMID),IY(JMID)+JAD
        WRITE(LU,103)IX(IMID),IX(IMID)+IAD,IY(JMID),IY(JMID)+JAD
        WRITE(LU,103)IX(IMID)+IAD,IX(IMID+1),IY(JMID),IY(JMID)+JAD
        WRITE(LU,103)IX(IMID-1),IX(IMID)-IAD,IY(JMID)+JAD,IY(JMID+1)
        WRITE(LU,103)IX(IMID)-IAD,IX(IMID),IY(JMID)+JAD,IY(JMID+1)
        WRITE(LU,103)IX(IMID),IX(IMID)+IAD,IY(JMID)+JAD,IY(JMID+1)
        WRITE(LU,103)IX(IMID)+IAD,IX(IMID+1),IY(JMID)+JAD,IY(JMID+1)
C
        CALL WRTFI2(LU,IRX,IRY,IX,IY,IFSR,JFSR)
        WRITE(LU,105)IX(1),IX(IRX+1),IY(1),IY(IRY+1)
        GOTO 561
      ENDIF
C
C-pd---Overwrite line info with arc data------------------------------
C-pd---Inner circle---------------------------------------------------
C
      ANG1=  0.0
      ANG2= 45.0
      ANG3= 90.0
      ANG4=135.0
      ANG5=180.0
      ANG6=225.0
      ANG7=270.0
      ANG8=315.0
      IFST=IG(117)+1
      JFST=IG(137)+1
      IMID=IFST+1
      JMID=JFST+1
      ILST=IFST+2
      JLST=JFST+2
      IBEF=IFST-1
      JBEF=JFST-1
      IAFT=IFST+3
      JAFT=JFST+3
      WRITE(LU,*)
      WRITE(LU,104)IX(IFST),IX(IMID),IY(JFST),IY(JFST),
     &              XCENB,YCENB,RAD1,ANG6,ANG7,XP(IFST)
      WRITE(LU,104)IX(IMID),IX(ILST),IY(JFST),IY(JFST),
     &              XCENB,YCENB,RAD1,ANG7,ANG8,XP(IMID)
      WRITE(LU,104)IX(IFST),IX(IMID),IY(JLST),IY(JLST),
     &              XCENB,YCENB,RAD1,ANG4,ANG3,XP(IFST)
      WRITE(LU,104)IX(IMID),IX(ILST),IY(JLST),IY(JLST),
     &              XCENB,YCENB,RAD1,ANG3,ANG2,XP(IMID)
      WRITE(LU,104)IX(IFST),IX(IFST),IY(JFST),IY(JMID),
     &              XCENB,YCENB,RAD1,ANG6,ANG5,YP(JFST)
      WRITE(LU,104)IX(IFST),IX(IFST),IY(JMID),IY(JLST),
     &              XCENB,YCENB,RAD1,ANG5,ANG4,YP(JMID)
      WRITE(LU,104)IX(ILST),IX(ILST),IY(JFST),IY(JMID),
     &              XCENB,YCENB,RAD1,ANG8,ANG1,YP(JFST)
      WRITE(LU,104)IX(ILST),IX(ILST),IY(JMID),IY(JLST),
     &              XCENB,YCENB,RAD1,ANG1,ANG2,YP(JMID)
C
C-pd---Shuffle lines--------------------------------------------------
C
      WRITE(LU,102)IX(IMID),IX(IMID),IY(JFST),IY(JMID),
     &              XCENB,YCENB-RAD1,XCENB,YCENB,YP(JFST)
      WRITE(LU,102)IX(IMID),IX(IMID),IY(JMID),IY(JLST),
     &              XCENB,YCENB,XCENB,YCENB+RAD1,YP(JMID)
      WRITE(LU,102)IX(IFST),IX(IMID),IY(JMID),IY(JMID),
     &              XCENB-RAD1,YCENB,XCENB,YCENB,XP(IFST)
      WRITE(LU,102)IX(IMID),IX(ILST),IY(JMID),IY(JMID),
     &              XCENB,YCENB,XCENB+RAD1,YCENB,XP(IMID)
C
C-pd---Outer circle---------------------------------------------------
C
      ANG01=  0.0
      ANG02=  0.0+TETT02
      ANG03= 45.0
      ANG04= 90.0-TETT04
      ANG05= 90.0
      ANG06= 90.0+TETT06
      ANG07=135.0
      ANG08=180.0-TETT08
      ANG09=180.0
      ANG10=180.0+TETT10
      ANG11=225.0
      ANG12=270.0-TETT12
      ANG13=270.0
      ANG14=270.0+TETT14
      ANG15=315.0
      ANG16=360.0-TETT16
      IFST=IG(117)
      JFST=IG(137)
      IMID=IFST+2
      JMID=JFST+2
      ILST=IFST+4
      JLST=JFST+4
      IBEF=IFST-1
      JBEF=JFST-1
      IAFT=IFST+5
      JAFT=JFST+5
      WRITE(LU,*)
      WRITE(LU,104)IX(IFST),IX(IFST+1),IY(JFST),IY(JFST),
     &              XCENA,YCENA,RAD2,ANG11,ANG12,XP(IFST)
      WRITE(LU,104)IX(IFST+1),IX(IMID),IY(JFST),IY(JFST),
     &              XCENA,YCENA,RAD2,ANG12,ANG13,XP(IFST+1)
      WRITE(LU,104)IX(IMID),IX(ILST-1),IY(JFST),IY(JFST),
     &              XCENA,YCENA,RAD2,ANG13,ANG14,XP(IMID)
      WRITE(LU,104)IX(ILST-1),IX(ILST),IY(JFST),IY(JFST),
     &              XCENA,YCENA,RAD2,ANG14,ANG15,XP(ILST-1)
      WRITE(LU,104)IX(IFST),IX(IFST+1),IY(JLST),IY(JLST),
     &              XCENA,YCENA,RAD2,ANG07,ANG06,XP(IFST)
      WRITE(LU,104)IX(IFST+1),IX(IMID),IY(JLST),IY(JLST),
     &              XCENA,YCENA,RAD2,ANG06,ANG05,XP(IFST+1)
      WRITE(LU,104)IX(IMID),IX(ILST-1),IY(JLST),IY(JLST),
     &              XCENA,YCENA,RAD2,ANG05,ANG04,XP(IMID)
      WRITE(LU,104)IX(ILST-1),IX(ILST),IY(JLST),IY(JLST),
     &              XCENA,YCENA,RAD2,ANG04,ANG03,XP(ILST-1)
      WRITE(LU,104)IX(IFST),IX(IFST),IY(JFST),IY(JFST+1),
     &              XCENA,YCENA,RAD2,ANG11,ANG10,YP(JFST)
      WRITE(LU,104)IX(IFST),IX(IFST),IY(JFST+1),IY(JMID),
     &              XCENA,YCENA,RAD2,ANG10,ANG09,YP(JFST+1)
      WRITE(LU,104)IX(IFST),IX(IFST),IY(JMID),IY(JLST-1),
     &              XCENA,YCENA,RAD2,ANG09,ANG08,YP(JMID)
      WRITE(LU,104)IX(IFST),IX(IFST),IY(JLST-1),IY(JLST),
     &              XCENA,YCENA,RAD2,ANG08,ANG07,YP(JLST-1)
      WRITE(LU,104)IX(ILST),IX(ILST),IY(JFST),IY(JFST+1),
     &              XCENA,YCENA,RAD2,ANG15,ANG16,YP(JFST)
      WRITE(LU,104)IX(ILST),IX(ILST),IY(JFST+1),IY(JMID),
     &              XCENA,YCENA,RAD2,ANG16,ANG01,YP(JFST+1)
      WRITE(LU,104)IX(ILST),IX(ILST),IY(JMID),IY(JLST-1),
     &              XCENA,YCENA,RAD2,ANG01,ANG02,YP(JMID)
      WRITE(LU,104)IX(ILST),IX(ILST),IY(JLST-1),IY(JLST),
     &              XCENA,YCENA,RAD2,ANG02,ANG03,YP(JLST-1)
C
C-pd---Shuffle lines-------------------------------------------------
C
      WRITE(LU,102)IX(IMID),IX(IMID),IY(JBEF),IY(JFST),
     &              XL(IMID),YL(JBEF),XCENA,YCENA-RAD2,YP(JBEF)
      WRITE(LU,102)IX(IMID),IX(IMID),IY(JFST),IY(JFST+1),
     &              XCENA,YCENA-RAD2,XCENB,YCENB-RAD1,YP(JFST)
      WRITE(LU,102)IX(IMID),IX(IMID),IY(JLST-1),IY(JLST),
     &              XCENB,YCENB+RAD1,XCENA,YCENA+RAD2,YP(JLST-1)
      WRITE(LU,102)IX(IMID),IX(IMID),IY(JLST),IY(JAFT),
     &              XCENA,YCENA+RAD2,XL(IMID),YL(JAFT),YP(JLST)
      WRITE(LU,102)IX(IBEF),IX(IFST),IY(JMID),IY(JMID),
     &              XL(IBEF),YL(JMID),XCENA-RAD2,YCENA,XP(IBEF)
      WRITE(LU,102)IX(IFST),IX(IFST+1),IY(JMID),IY(JMID),
     &              XCENA-RAD2,YCENA,XCENB-RAD1,YCENB,XP(IFST)
      WRITE(LU,102)IX(ILST-1),IX(ILST),IY(JMID),IY(JMID),
     &              XCENB+RAD1,YCENB,XCENA+RAD2,YCENA,XP(ILST-1)
      WRITE(LU,102)IX(ILST),IX(IAFT),IY(JMID),IY(JMID),
     &              XCENA+RAD2,YCENA,XL(IAFT),YL(JMID),XP(ILST)
C
C-pd---More trig-----------------------------------------------------
C
      DELL02=(RAD2*RAD2-DXI02*DXI02)**0.5
      DELL04=(RAD2*RAD2-DXI04*DXI04)**0.5
      DELL06=(RAD2*RAD2-DXI06*DXI06)**0.5
      DELL08=(RAD2*RAD2-DXI08*DXI08)**0.5
      DELL10=(RAD2*RAD2-DXI10*DXI10)**0.5
      DELL12=(RAD2*RAD2-DXI12*DXI12)**0.5
      DELL14=(RAD2*RAD2-DXI14*DXI14)**0.5
      DELL16=(RAD2*RAD2-DXI16*DXI16)**0.5
C
      WRITE(LU,*)
      WRITE(LU,102)IX(IFST+1),IX(IFST+1),IY(JBEF),IY(JFST),
     &          XL(IFST+1),YL(JBEF),XL(IFST+1),YCENA-DELL12,YP(JBEF)
      WRITE(LU,102)IX(IFST+1),IX(IFST+1),IY(JFST),IY(JFST+1),
     &          XL(IFST+1),YCENA-DELL12,XL(IFST+1),YL(JFST+1),YP(JFST)
      WRITE(LU,102)IX(IFST+1),IX(IFST+1),IY(JLST-1),IY(JLST),
     &          XL(IFST+1),YL(JLST-1),XL(IFST+1),YCENA+DELL06,YP(JLST-1)
      WRITE(LU,102)IX(IFST+1),IX(IFST+1),IY(JLST),IY(JAFT),
     &          XL(IFST+1),YCENA+DELL06,XL(IFST+1),YL(JAFT),YP(JLST)
      WRITE(LU,102)IX(ILST-1),IX(ILST-1),IY(JBEF),IY(JFST),
     &          XL(ILST-1),YL(JBEF),XL(ILST-1),YCENA-DELL14,YP(JBEF)
      WRITE(LU,102)IX(ILST-1),IX(ILST-1),IY(JFST),IY(JFST+1),
     &          XL(ILST-1),YCENA-DELL14,XL(ILST-1),YL(JFST+1),YP(JFST)
      WRITE(LU,102)IX(ILST-1),IX(ILST-1),IY(JLST-1),IY(JLST),
     &          XL(ILST-1),YL(JLST-1),XL(ILST-1),YCENA+DELL04,YP(JLST-1)
      WRITE(LU,102)IX(ILST-1),IX(ILST-1),IY(JLST),IY(JAFT),
     &          XL(ILST-1),YCENA+DELL04,XL(ILST-1),YL(JAFT),YP(JLST)
C
      WRITE(LU,102)IX(IBEF),IX(IFST),IY(JFST+1),IY(JFST+1),
     &          XL(IBEF),YL(JFST+1),XCENA-DELL10,YL(JFST+1),XP(IBEF)
      WRITE(LU,102)IX(IFST),IX(IFST+1),IY(JFST+1),IY(JFST+1),
     &          XCENA-DELL10,YL(JFST+1),XL(IFST+1),YL(JFST+1),XP(IFST)
      WRITE(LU,102)IX(ILST-1),IX(ILST),IY(JFST+1),IY(JFST+1),
     &          XL(ILST-1),YL(JFST+1),XCENA+DELL16,YL(JFST+1),XP(ILST-1)
      WRITE(LU,102)IX(ILST),IX(IAFT),IY(JFST+1),IY(JFST+1),
     &          XCENA+DELL16,YL(JFST+1),XL(IAFT),YL(JFST+1),XP(ILST)
      WRITE(LU,102)IX(IBEF),IX(IFST),IY(JLST-1),IY(JLST-1),
     &          XL(IBEF),YL(JLST-1),XCENA-DELL08,YL(JLST-1),XP(IBEF)
      WRITE(LU,102)IX(IFST),IX(IFST+1),IY(JLST-1),IY(JLST-1),
     &          XCENA-DELL08,YL(JLST-1),XL(IFST+1),YL(JLST-1),XP(IFST)
      WRITE(LU,102)IX(ILST-1),IX(ILST),IY(JLST-1),IY(JLST-1),
     &          XL(ILST-1),YL(JLST-1),XCENA+DELL02,YL(JLST-1),XP(ILST-1)
      WRITE(LU,102)IX(ILST),IX(IAFT),IY(JLST-1),IY(JLST-1),
     &          XCENA+DELL02,YL(JLST-1),XL(IAFT),YL(JLST-1),XP(ILST)
      WRITE(LU,*)
C
C-pd---Add lines for upper gap------------------------------------
      IF(I.EQ.2.OR.I.EQ.3) THEN
        IADD=IG(61)/2
        XDST=RG(61)/2.
        XTP=1.0
        WRITE(LU,102)IX(IMID)-IADD,IX(IMID)-IADD,IY(JAFT),IY(IRY),
     &            XCENA-XDST,YL(JAFT),XCENA-XDST,YL(IRY),XTP
        WRITE(LU,102)IX(IMID)-IADD,IX(IMID)-IADD,IY(IRY),IY(IRY+1),
     &            XCENA-XDST,YL(IRY),XCENA-XDST,YL(IRY+1),XTP
        WRITE(LU,102)IX(IMID)+IADD,IX(IMID)+IADD,IY(JAFT),IY(IRY),
     &            XCENA+XDST,YL(JAFT),XCENA+XDST,YL(IRY),XTP
        WRITE(LU,102)IX(IMID)+IADD,IX(IMID)+IADD,IY(IRY),IY(IRY+1),
     &            XCENA+XDST,YL(IRY),XCENA+XDST,YL(IRY+1),XTP
        WRITE(LU,102)IX(IMID)-IADD,IX(IMID)+IADD,IY(JAFT),IY(JAFT),
     &            XCENA-XDST,YL(JAFT),XCENA+XDST,YL(JAFT),XTP
        WRITE(LU,102)IX(IMID)-IADD,IX(IMID)+IADD,IY(IRY),IY(IRY),
     &            XCENA-XDST,YL(IRY),XCENA+XDST,YL(IRY),XTP
        WRITE(LU,102)IX(IMID)-IADD,IX(IMID)+IADD,IY(IRY+1),IY(IRY+1),
     &            XCENA-XDST,YL(IRY+1),XCENA+XDST,YL(IRY+1),XTP
      ENDIF
C
```

```
C-pd---Overwrite line info with arc data-----------------------------
C
      IFSR=IG(117)+1
      JFSR=IG(137)+1
      IMID=IFSR+1
      JMID=JFSR+1
      IAD=IG(50)
      JAD=IG(51)
      RAD3=RG(53)/2.
      IF(I.EQ.5) THEN
        XCENB=RG(45)
        YCENB=RG(46)
        RAD3=RG(54)/2.
      ENDIF
C
      WRITE(LU,*)
      WRITE(LU,104)IX(IMID)-IAD,IX(IMID),IY(JMID)-JAD,IY(JMID)-JAD,
     &             XCENB,YCENB,RAD3,ANG6,ANG7,1.0
      WRITE(LU,104)IX(IMID),IX(IMID)+IAD,IY(JMID)-JAD,IY(JMID)-JAD,
     &             XCENB,YCENB,RAD3,ANG7,ANG8,1.0
      WRITE(LU,104)IX(IMID)-IAD,IX(IMID),IY(JMID)+JAD,IY(JMID)+JAD,
     &             XCENB,YCENB,RAD3,ANG4,ANG3,1.0
      WRITE(LU,104)IX(IMID),IX(IMID)+IAD,IY(JMID)+JAD,IY(JMID)+JAD,
     &             XCENB,YCENB,RAD3,ANG3,ANG2,1.0
      WRITE(LU,104)IX(IMID)-IAD,IX(IMID)-IAD,IY(JMID)-JAD,IY(JMID),
     &             XCENB,YCENB,RAD3,ANG6,ANG5,1.0
      WRITE(LU,104)IX(IMID)-IAD,IX(IMID)-IAD,IY(JMID),IY(JMID)+JAD,
     &             XCENB,YCENB,RAD3,ANG5,ANG4,1.0
      WRITE(LU,104)IX(IMID)+IAD,IX(IMID)+IAD,IY(JMID)-JAD,IY(JMID),
     &             XCENB,YCENB,RAD3,ANG8,ANG1,1.0
      WRITE(LU,104)IX(IMID)+IAD,IX(IMID)+IAD,IY(JMID),IY(JMID)+JAD,
     &             XCENB,YCENB,RAD3,ANG1,ANG2,1.0
C
C-pd---Shuffle lines-------------------------------------------------
C
      XCENC=XCENB
      YCENC=YCENB
      IF(I.EQ.5) THEN
        XCENC=RG(43)
        YCENC=RG(44)
      ENDIF
      WRITE(LU,*)
      WRITE(LU,102)IX(IMID-1),IX(IMID)-IAD,IY(JMID),IY(JMID),
     &             XCENC-RAD1,YCENC,XCENB-RAD3,YCENB,1.0
      WRITE(LU,102)IX(IMID)-IAD,IX(IMID),IY(JMID),IY(JMID),
     &             XCENB-RAD3,YCENB,XCENB,YCENB,1.0
      WRITE(LU,102)IX(IMID),IX(IMID)+IAD,IY(JMID),IY(JMID),
     &             XCENB,YCENB,XCENB+RAD3,YCENB,1.0
      WRITE(LU,102)IX(IMID)+IAD,IX(IMID+1),IY(JMID),IY(JMID),
     &             XCENB+RAD3,YCENB,XCENC+RAD1,YCENC,1.0
      WRITE(LU,102)IX(IMID),IX(IMID),IY(JMID-1),IY(JMID)-JAD,
     &             XCENC,YCENC-RAD1,XCENB,YCENB-RAD3,1.0
      WRITE(LU,102)IX(IMID),IX(IMID),IY(JMID)-JAD,IY(JMID),
     &             XCENB,YCENB-RAD3,XCENB,YCENB,1.0
      WRITE(LU,102)IX(IMID),IX(IMID),IY(JMID),IY(JMID)+JAD,
     &             XCENB,YCENB,XCENB,YCENB+RAD3,1.0
      WRITE(LU,102)IX(IMID),IX(IMID),IY(JMID)+JAD,IY(JMID+1),
     &             XCENB,YCENB+RAD3,XCENC,YCENC+RAD1,1.0
C
      WRITE(LU,*)
      WRITE(LU,103)IX(IMID-1),IX(IMID)-IAD,IY(JMID-1),IY(JMID)-JAD
      WRITE(LU,103)IX(IMID)-IAD,IX(IMID),IY(JMID-1),IY(JMID)-JAD
      WRITE(LU,103)IX(IMID),IX(IMID)+IAD,IY(JMID-1),IY(JMID)-JAD
      WRITE(LU,103)IX(IMID)+IAD,IX(IMID+1),IY(JMID-1),IY(JMID)-JAD
      WRITE(LU,103)IX(IMID-1),IX(IMID)-IAD,IY(JMID)-JAD,IY(JMID)
      WRITE(LU,103)IX(IMID)-IAD,IX(IMID),IY(JMID)-JAD,IY(JMID)
      WRITE(LU,103)IX(IMID),IX(IMID)+IAD,IY(JMID)-JAD,IY(JMID)
      WRITE(LU,103)IX(IMID)+IAD,IX(IMID+1),IY(JMID)-JAD,IY(JMID)
      WRITE(LU,103)IX(IMID-1),IX(IMID)-IAD,IY(JMID),IY(JMID)+JAD
      WRITE(LU,103)IX(IMID)-IAD,IX(IMID),IY(JMID),IY(JMID)+JAD
      WRITE(LU,103)IX(IMID),IX(IMID)+IAD,IY(JMID),IY(JMID)+JAD
      WRITE(LU,103)IX(IMID)+IAD,IX(IMID+1),IY(JMID),IY(JMID)+JAD
      WRITE(LU,103)IX(IMID-1),IX(IMID)-IAD,IY(JMID)+JAD,IY(JMID+1)
      WRITE(LU,103)IX(IMID)-IAD,IX(IMID),IY(JMID)+JAD,IY(JMID+1)
      WRITE(LU,103)IX(IMID),IX(IMID)+IAD,IY(JMID)+JAD,IY(JMID+1)
      WRITE(LU,103)IX(IMID)+IAD,IX(IMID+1),IY(JMID)+JAD,IY(JMID+1)
C
      CALL WRTFI2(LU,IRX,IRY,IX,IY,IFSR,JFSR)
C
C-pd---Fix points around circle and certain ones inside--------------
C
      WRITE(LU,*)
      WRITE(LU,105)IX(1),IX(IRX+1),IY(1),IY(JFST)
      WRITE(LU,105)IX(1),IX(IFST),IY(JFST),IY(JLST)
      WRITE(LU,105)IX(ILST),IX(IRX+1),IY(JFST),IY(JLST)
      WRITE(LU,105)IX(1),IX(IRX+1),IY(JLST),IY(IRY+1)
C
      ISOL=3
      WRITE(LU,105)IX(IFST+1)+ISOL,IX(ILST-1)-ISOL,IY(JFST+1),IY(JLST-1)
      WRITE(LU,105)IX(IFST+1),IX(ILST-1),IY(JFST+1)+ISOL,IY(JLST-1)-ISOL
C
      WRITE(LU,105)IX(IFST+1),IX(ILST-1),IY(JFST),IY(JFST+1)
      WRITE(LU,105)IX(IFST),IX(IFST+1),IY(JFST+1),IY(JLST-1)
      WRITE(LU,105)IX(ILST-1),IX(ILST),IY(JFST+1),IY(JLST-1)
      WRITE(LU,105)IX(IFST+1),IX(ILST-1),IY(JLST-1),IY(JLST)
C
  561 CONTINUE
C
C*********************************************************************
C-pd---This is the second option for exit of the engine.  There will--
C---   will be either 2 or 3 cross sections written here depending----
C---   on the location exit.  It will be 2 if it ends before the------
C---   augmenter tube or if it ends at the start of the tappered------
C---   section or at the start of straight section.  It will be 3-----
C---   if it falls in the tappered section or after the start of------
C---   the straight section.-----------------------------------------
C    t3
      DO 563 I=1,IG(60)
C
      IX(1 )=1
      CALL SETIV(IX,IG,140,1,NI)
C
      XCENC=RG(47)
      YCENC=RG(48)
      XCEND=RG(47)
      YCEND=RG(48)
      RAD1=RG(54)/2.
      DXI=(RAD1*RAD1/2.)**0.5
      RAD2=RG(54+I)/2.
      DXII=(RAD2*RAD2/2.)**0.5
      IFST=IG(157)
      JFST=IG(177)
C
C-pd---Do trig-------------------------------------------------------
C
      DXI02=DXI+(YCENC-YCEND)
      TETT02=ASIN(DXI02/RAD2)*180./3.141592654
      DXI16=DXI-(YCENC-YCEND)
      TETT16=ASIN(DXI16/RAD2)*180./3.141592654
      DXI04=DXI+(XCENC-XCEND)
      TETT04=ASIN(DXI04/RAD2)*180./3.141592654
      DXI06=DXI-(XCENC-XCEND)
      TETT06=ASIN(DXI06/RAD2)*180./3.141592654
      DXI08=DXI+(YCENC-YCEND)
      TETT08=ASIN(DXI08/RAD2)*180./3.141592654
      DXI10=DXI-(YCENC-YCEND)
      TETT10=ASIN(DXI10/RAD2)*180./3.141592654
      DXI12=DXI-(XCENC-XCEND)
      TETT12=ASIN(DXI12/RAD2)*180./3.141592654
      DXI14=DXI+(XCENC-XCEND)
      TETT14=ASIN(DXI14/RAD2)*180./3.141592654
C
      IRX=IG(44)
      IRY=IG(45)
C
      XL(1 )=0.0
      CALL SETRV(XL,RG,180,1,NI)
      XL(IFST+1)=XCEND-DXI
      XL(IFST+3)=XCEND+DXI
      XL(IFST  )=XCEND-DXII
      XL(IFST+4)=XCEND+DXII
      XL(IFST-1)=XCEND-((XCEND-RAD2)/2.)-RAD2
      XL(IFST+5)=XCEND+((XL(IRX+1)-XCEND-RAD2)/2.)+RAD2
C
      CALL SETRV(XP,RG,200,2,NI)
C
      IY(1 )=1
      CALL SETIV(IY,IG,160,1,NI)
C
      YL(1 )=0.0
      CALL SETRV(YL,RG,220,1,NI)
      YL(JFST+1)=YCEND-DXI
      YL(JFST+3)=YCEND+DXI
      YL(JFST  )=YCEND-DXII
      YL(JFST+4)=YCEND+DXII
      YL(JFST-1)=YCEND-((YCEND-RAD2)/2.)-RAD2
      YL(JFST+5)=YCEND+((YL(IRY+1)-YCEND-RAD2)/2.)+RAD2
C
      CALL SETRV(YP,RG,240,2,NI)
C
      LU=65+I
      CG(LU)='CS  '
      I10=LU/10
      I1=LU-I10*10
      WRITE(CG(LU)(3:3),'(I1)') I10
      WRITE(CG(LU)(4:4),'(I1)') I1
      OPEN(LU,FILE=CG(LU),FORM='FORMATTED',STATUS='UNKNOWN')
      IF(RG(LU+10).NE.0.0) XL(1)=RG(LU+10)
      CALL WRTSQ(LU,NX,NY,IRX,IRY,IX,IY,XL,YL,XP,YP)
C
```

```
C-pd---Overwrite line info with arc data------------------------------
C-pd---Inner circle--------------------------------------------------

      XCENC=RG(45)
      YCENC=RG(46)

      ANG1=  0.0
      ANG2= 45.0
      ANG3= 90.0
      ANG4=135.0
      ANG5=180.0
      ANG6=225.0
      ANG7=270.0
      ANG8=315.0
      IFST=IG(157)+1
      JFST=IG(177)+1
      IMID=IFST+1
      JMID=JFST+1
      ILST=IFST+2
      JLST=JFST+2
      IBEF=IFST-1
      JBEF=JFST-1
      IAFT=IFST+3
      JAFT=JFST+3
      WRITE(LU,*)
      WRITE(LU,104)IX(IFST),IX(IMID),IY(JFST),IY(JFST),
     &              XCENC,YCENC,RAD1,ANG6,ANG7,XP(IFST)
      WRITE(LU,104)IX(IMID),IX(ILST),IY(JFST),IY(JFST),
     &              XCENC,YCENC,RAD1,ANG7,ANG8,XP(IMID)
      WRITE(LU,104)IX(IFST),IX(IMID),IY(JLST),IY(JLST),
     &              XCENC,YCENC,RAD1,ANG4,ANG3,XP(IFST)
      WRITE(LU,104)IX(IMID),IX(ILST),IY(JLST),IY(JLST),
     &              XCENC,YCENC,RAD1,ANG3,ANG2,XP(IMID)
      WRITE(LU,104)IX(IFST),IX(IFST),IY(JFST),IY(JMID),
     &              XCENC,YCENC,RAD1,ANG6,ANG5,YP(JFST)
      WRITE(LU,104)IX(IFST),IX(IFST),IY(JMID),IY(JLST),
     &              XCENC,YCENC,RAD1,ANG5,ANG4,YP(JMID)
      WRITE(LU,104)IX(ILST),IX(ILST),IY(JFST),IY(JMID),
     &              XCENC,YCENC,RAD1,ANG8,ANG1,YP(JFST)
      WRITE(LU,104)IX(ILST),IX(ILST),IY(JMID),IY(JLST),
     &              XCENC,YCENC,RAD1,ANG1,ANG2,YP(JMID)
C
C-pd---Shuffle lines-------------------------------------------------
C
      WRITE(LU,102)IX(IMID),IX(IMID),IY(JFST),IY(JMID),
     &              XCENC,YCENC-RAD1,XCENC,YCENC,YP(JFST)
      WRITE(LU,102)IX(IMID),IX(IMID),IY(JMID),IY(JLST),
     &              XCENC,YCENC,XCENC,YCENC+RAD1,YP(JMID)
      WRITE(LU,102)IX(IFST),IX(IMID),IY(JMID),IY(JMID),
     &              XCENC-RAD1,YCENC,XCENC,YCENC,XP(IFST)
      WRITE(LU,102)IX(IMID),IX(ILST),IY(JMID),IY(JMID),
     &              XCENC,YCENC,XCENC+RAD1,YCENC,XP(IMID)
C
C-pd---Outer circle--------------------------------------------------
C
      ANGO1=  0.0
      ANGO2=  0.0+TETTO2
      ANGO3= 45.0
      ANGO4= 90.0-TETTO4
      ANGO5= 90.0
      ANG06= 90.0+TETT06
      ANG07=135.0
      ANG08=180.0-TETT08
      ANG09=180.0
      ANG10=180.0+TETT10
      ANG11=225.0
      ANG12=270.0-TETT12
      ANG13=270.0
      ANG14=270.0+TETT14
      ANG15=315.0
      ANG16=360.0-TETT16
      IFST=IG(157)
      JFST=IG(177)
      IMID=IFST+2
      JMID=JFST+2
      ILST=IFST+4
      JLST=JFST+4
      IBEF=IFST-1
      JBEF=JFST-1
      IAFT=IFST+5
      JAFT=JFST+5
      WRITE(LU,*)
      WRITE(LU,104)IX(IFST),IX(IFST+1),IY(JFST),IY(JFST),
     &              XCEND,YCEND,RAD2,ANG11,ANG12,XP(IFST)
      WRITE(LU,104)IX(IFST+1),IX(IMID),IY(JFST),IY(JFST),
     &              XCEND,YCEND,RAD2,ANG12,ANG13,XP(IFST+1)
      WRITE(LU,104)IX(IMID),IX(ILST-1),IY(JFST),IY(JFST),
     &              XCEND,YCEND,RAD2,ANG13,ANG14,XP(IMID)
      WRITE(LU,104)IX(ILST-1),IX(ILST),IY(JFST),IY(JFST),
     &              XCEND,YCEND,RAD2,ANG14,ANG15,XP(ILST-1)
      WRITE(LU,104)IX(IFST),IX(IFST+1),IY(JLST),IY(JLST),
     &              XCEND,YCEND,RAD2,ANG07,ANG06,XP(IFST)
      WRITE(LU,104)IX(IFST+1),IX(IMID),IY(JLST),IY(JLST),
     &              XCEND,YCEND,RAD2,ANG06,ANG05,XP(IFST+1)
      WRITE(LU,104)IX(IMID),IX(ILST-1),IY(JLST),IY(JLST),
     &              XCEND,YCEND,RAD2,ANG05,ANG04,XP(IMID)
      WRITE(LU,104)IX(ILST-1),IX(ILST),IY(JLST),IY(JLST),
     &              XCEND,YCEND,RAD2,ANG04,ANG03,XP(ILST-1)
      WRITE(LU,104)IX(IFST),IX(IFST),IY(JFST),IY(JFST+1),
     &              XCEND,YCEND,RAD2,ANG11,ANG10,YP(JFST)
      WRITE(LU,104)IX(IFST),IX(IFST),IY(JFST+1),IY(JMID),
     &              XCEND,YCEND,RAD2,ANG10,ANG09,YP(JFST+1)
      WRITE(LU,104)IX(IFST),IX(IFST),IY(JMID),IY(JLST-1),
     &              XCEND,YCEND,RAD2,ANG09,ANG08,YP(JMID)
      WRITE(LU,104)IX(IFST),IX(IFST),IY(JLST-1),IY(JLST),
     &              XCEND,YCEND,RAD2,ANG08,ANG07,YP(JLST-1)
      WRITE(LU,104)IX(ILST),IX(ILST),IY(JFST),IY(JFST+1),
     &              XCEND,YCEND,RAD2,ANG15,ANG16,YP(JFST)
      WRITE(LU,104)IX(ILST),IX(ILST),IY(JFST+1),IY(JMID),
     &              XCEND,YCEND,RAD2,ANG16,ANG01,YP(JFST+1)
      WRITE(LU,104)IX(ILST),IX(ILST),IY(JMID),IY(JLST-1),
     &              XCEND,YCEND,RAD2,ANG01,ANG02,YP(JMID)
      WRITE(LU,104)IX(ILST),IX(ILST),IY(JLST-1),IY(JLST),
     &              XCEND,YCEND,RAD2,ANG02,ANG03,YP(JLST-1)
C
C-pd---Shuffle lines------------------------------------------------
C
      WRITE(LU,102)IX(IMID),IX(IMID),IY(JBEF),IY(JFST),
     &              XL(IMID),YL(JBEF),XCEND,YCEND-RAD2,YP(JBEF)
      WRITE(LU,102)IX(IMID),IX(IMID),IY(JFST),IY(JFST+1),
     &              XCEND,YCEND-RAD2,XCENC,YCENC-RAD1,YP(JFST)
      WRITE(LU,102)IX(IMID),IX(IMID),IY(JLST-1),IY(JLST),
     &              XCENC,YCENC+RAD1,XCEND,YCEND+RAD2,YP(JLST-1)
      WRITE(LU,102)IX(IMID),IX(IMID),IY(JLST),IY(JAFT),
     &              XCEND,YCEND+RAD2,XL(IMID),YL(JAFT),YP(JLST)
      WRITE(LU,102)IX(IBEF),IX(IFST),IY(JMID),IY(JMID),
     &              XL(IBEF),YL(JMID),XCEND-RAD2,YCEND,XP(IBEF)
      WRITE(LU,102)IX(IFST),IX(IFST+1),IY(JMID),IY(JMID),
     &              XCEND-RAD2,YCEND,XCENC-RAD1,YCENC,XP(IFST)
      WRITE(LU,102)IX(ILST-1),IX(ILST),IY(JMID),IY(JMID),
     &              XCENC+RAD1,YCENC,XCEND+RAD2,YCEND,XP(ILST-1)
      WRITE(LU,102)IX(ILST),IX(IAFT),IY(JMID),IY(JMID),
     &              XCEND+RAD2,YCEND,XL(IAFT),YL(JMID),XP(ILST)

C-pd---More trig------------------------------------------------------

      DELL02=(RAD2*RAD2-DXI02*DXI02)**0.5
      DELL04=(RAD2*RAD2-DXI04*DXI04)**0.5
      DELL06=(RAD2*RAD2-DXI06*DXI06)**0.5
      DELL08=(RAD2*RAD2-DXI08*DXI08)**0.5
      DELL10=(RAD2*RAD2-DXI10*DXI10)**0.5
      DELL12=(RAD2*RAD2-DXI12*DXI12)**0.5
      DELL14=(RAD2*RAD2-DXI14*DXI14)**0.5
      DELL16=(RAD2*RAD2-DXI16*DXI16)**0.5
C
      WRITE(LU,*)
      WRITE(LU,102)IX(IFST+1),IX(IFST+1),IY(JBEF),IY(JFST),
     &          XL(IFST+1),YL(JBEF),XL(IFST+1),YCEND-DELL12,YP(JBEF)
      WRITE(LU,102)IX(IFST+1),IX(IFST+1),IY(JFST),IY(JFST+1),
     &          XL(IFST+1),YCEND-DELL12,XCENC-DXI,YCENC-DXI,YP(JFST)
      WRITE(LU,102)IX(IFST+1),IX(IFST+1),IY(JLST-1),IY(JLST),
     &          XCENC-DXI,YCENC+DXI,XL(IFST+1),YCEND+DELL06,YP(JLST-1)
      WRITE(LU,102)IX(IFST+1),IX(IFST+1),IY(JLST),IY(JAFT),
     &          XL(IFST+1),YCEND+DELL06,XL(IFST+1),YL(JAFT),YP(JLST)
      WRITE(LU,102)IX(ILST-1),IX(ILST-1),IY(JBEF),IY(JFST),
     &          XL(ILST-1),YL(JBEF),XL(ILST-1),YCEND-DELL14,YP(JBEF)
      WRITE(LU,102)IX(ILST-1),IX(ILST-1),IY(JFST),IY(JFST+1),
     &          XL(ILST-1),YCEND-DELL14,XCENC+DXI,YCENC-DXI,YP(JFST)
      WRITE(LU,102)IX(ILST-1),IX(ILST-1),IY(JLST-1),IY(JLST),
     &          XCENC+DXI,YCENC+DXI,XL(ILST-1),YCEND+DELL04,YP(JLST-1)
      WRITE(LU,102)IX(ILST-1),IX(ILST-1),IY(JLST),IY(JAFT),
     &          XL(ILST-1),YCEND+DELL04,XL(ILST-1),YL(JAFT),YP(JLST)

      WRITE(LU,102)IX(IBEF),IX(IFST),IY(JFST+1),IY(JFST+1),
     &          XL(IBEF),YL(JFST+1),XCEND-DELL10,YL(JFST+1),XP(IBEF)
      WRITE(LU,102)IX(IFST),IX(IFST+1),IY(JFST+1),IY(JFST+1),
     &          XCEND-DELL10,YL(JFST+1),XCENC-DXI,YCENC-DXI,XP(IFST)
      WRITE(LU,102)IX(ILST-1),IX(ILST),IY(JFST+1),IY(JFST+1),
     &          XCENC+DXI,YCENC-DXI,XCEND+DELL16,YL(JFST+1),XP(ILST-1)
      WRITE(LU,102)IX(ILST),IX(IAFT),IY(JFST+1),IY(JFST+1),
     &          XCEND+DELL16,YL(JFST+1),XL(IAFT),YL(JFST+1),XP(ILST)
      WRITE(LU,102)IX(IBEF),IX(IFST),IY(JLST-1),IY(JLST-1),
     &          XL(IBEF),YL(JLST-1),XCEND-DELL08,YL(JLST-1),XP(IBEF)
      WRITE(LU,102)IX(IFST),IX(IFST+1),IY(JLST-1),IY(JLST-1),
     &          XCEND-DELL08,YL(JLST-1),XCENC-DXI,YCENC+DXI,XP(IFST)
      WRITE(LU,102)IX(ILST-1),IX(ILST),IY(JLST-1),IY(JLST-1),
     &          XCENC+DXI,YCENC+DXI,XCEND+DELL02,YL(JLST-1),XP(ILST-1)
      WRITE(LU,102)IX(ILST),IX(IAFT),IY(JLST-1),IY(JLST-1),
     &          XCEND+DELL02,YL(JLST-1),XL(IAFT),YL(JLST-1),XP(ILST)
```

```
      CALL WRTFI(LU,IRX,IRY,IX,IY)
C
C-pd---Fix points around circle and certain ones inside--------------
C
      WRITE(LU,*)
      WRITE(LU,105)IX(1),IX(IRX+1),IY(1),IY(JFST)
      WRITE(LU,105)IX(1),IX(IFST),IY(JFST),IY(JLST)
      WRITE(LU,105)IX(ILST),IX(IRX+1),IY(JFST),IY(JLST)
      WRITE(LU,105)IX(1),IX(IRX+1),IY(JLST),IY(IRY+1)
C
      ISOL=2
      WRITE(LU,105)IX(IFST+1)+ISOL,IX(ILST-1)-ISOL,IY(JFST+1),IY(JLST-1)
      WRITE(LU,105)IX(IFST+1),IX(ILST-1),IY(JFST+1)+ISOL,IY(JLST-1)-ISOL
C
      WRITE(LU,105)IX(IFST+1),IX(ILST-1),IY(JFST),IY(JFST+1)
      WRITE(LU,105)IX(IFST),IX(IFST+1),IY(JFST+1),IY(JLST-1)
      WRITE(LU,105)IX(ILST-1),IX(ILST),IY(JFST+1),IY(JLST-1)
      WRITE(LU,105)IX(IFST+1),IX(ILST-1),IY(JLST-1),IY(JLST)
C
  563 CONTINUE
C
C*********************************************************************
C-pd---This section is for the constant cross sectional area of the---
C---   augmenter sleeve.-------------------------------------------
C    t4
      DO 564 I=1,3
C
      IX(1 )=1
      CALL SETIV(IX,IG,180,1,NI)
C
      XCEND=RG(47)
      YCEND=RG(48)
      RAD1=RG(56+I)/2.
      IF(IG(60).EQ.2.AND.I.EQ.1) RAD1=RG(56)/2.
      DXI=(RAD1*RAD1/2.)**0.5
      IFST=IG(197)
      JFST=IG(217)
      IMID=IFST+1
      JMID=JFST+1
      ILST=IFST+2
      JLST=JFST+2
      IBEF=IFST-1
      JBEF=JFST-1
      IAFT=IFST+3
      JAFT=JFST+3
C
      IRX=IG(46)
      IRY=IG(47)
C
      XL(1 )=0.0
      CALL SETRV(XL,RG,260,1,NI)
      XL(IFST  )=XCEND-DXI
      XL(IFST+2)=XCEND+DXI
      XL(IFST-1)=XCEND-((XCEND-RAD1)/2.)-RAD1
      XL(IFST+3)=XCEND+((XL(IRX+1)-XCEND-RAD1)/2.)+RAD1
C
      CALL SETRV(XP,RG,280,2,NI)
C
      IY(1 )=1
      CALL SETIV(IY,IG,200,1,NI)
```

```
C
      YL(1 )=0.0
      CALL SETRV(YL,RG,300,1,NI)
      YL(JFST  )=YCEND-DXI
      YL(JFST+2)=YCEND+DXI
      YL(JFST-1)=YCEND-((YCEND-RAD1)/2.)-RAD1
      YL(JFST+3)=YCEND+((YL(IRY+1)-YCEND-RAD1)/2.)+RAD1
C
      CALL SETRV(YP,RG,320,2,NI)
C
      LU=65+IG(60)+I
      CG(LU)='CS  '
      I10=LU/10
      I1=LU-I10*10
      WRITE(CG(LU)(3:3),'(I1)') I10
      WRITE(CG(LU)(4:4),'(I1)') I1
      OPEN(LU,FILE=CG(LU),FORM='FORMATTED',STATUS='UNKNOWN')
      IF(RG(LU+10).NE.0.0) XL(1)=RG(LU+10)
      CALL WRTSQ(LU,NX,NY,IRX,IRY,IX,IY,XL,YL,XP,YP)
C
C-pd---Overwrite line info with arc data-----------------------------
C
      ANG1=  0.0
      ANG2= 45.0
      ANG3= 90.0
      ANG4=135.0
      ANG5=180.0
      ANG6=225.0
      ANG7=270.0
      ANG8=315.0
      WRITE(LU,*)
      WRITE(LU,104)IX(IFST),IX(IMID),IY(JFST),IY(JFST),
     &         XCEND,YCEND,RAD1,ANG6,ANG7,XP(IFST)
      WRITE(LU,104)IX(IMID),IX(ILST),IY(JFST),IY(JFST),
     &         XCEND,YCEND,RAD1,ANG7,ANG8,XP(IMID)
      WRITE(LU,104)IX(IFST),IX(IMID),IY(JLST),IY(JLST),
     &         XCEND,YCEND,RAD1,ANG4,ANG3,XP(IFST)
      WRITE(LU,104)IX(IMID),IX(ILST),IY(JLST),IY(JLST),
     &         XCEND,YCEND,RAD1,ANG3,ANG2,XP(IMID)
      WRITE(LU,104)IX(IFST),IX(IFST),IY(JFST),IY(JMID),
     &         XCEND,YCEND,RAD1,ANG6,ANG5,YP(JFST)
      WRITE(LU,104)IX(IFST),IX(IFST),IY(JMID),IY(JLST),
     &         XCEND,YCEND,RAD1,ANG5,ANG4,YP(JMID)
      WRITE(LU,104)IX(ILST),IX(ILST),IY(JFST),IY(JMID),
     &         XCEND,YCEND,RAD1,ANG8,ANG1,YP(JFST)
      WRITE(LU,104)IX(ILST),IX(ILST),IY(JMID),IY(JLST),
     &         XCEND,YCEND,RAD1,ANG1,ANG2,YP(JMID)
C
C-pd---Shuffle lines-------------------------------------------------
C
      WRITE(LU,102)IX(IMID),IX(IMID),IY(JBEF),IY(JFST),
     &         XL(IMID),YL(JBEF),XL(IMID),YCEND-RAD1,YP(JBEF)
      WRITE(LU,102)IX(IMID),IX(IMID),IY(JFST),IY(JMID),
     &         XL(IMID),YCEND-RAD1,XL(IMID),YCEND,YP(JFST)
      WRITE(LU,102)IX(IMID),IX(IMID),IY(JMID),IY(JLST),
     &         XL(IMID),YCEND,XL(IMID),YCEND+RAD1,YP(JMID)
      WRITE(LU,102)IX(IMID),IX(IMID),IY(JLST),IY(JAFT),
     &         XL(IMID),YCEND+RAD1,XL(IMID),YL(JAFT),YP(JLST)
      WRITE(LU,102)IX(IBEF),IX(IFST),IY(JMID),IY(JMID),
     &         XL(IBEF),YL(JMID),XCEND-RAD1,YL(JMID),XP(IBEF)
```

```
      WRITE(LU,102)IX(IFST),IX(IMID),IY(JMID),IY(JMID),
     &              XCEND-RAD1,YL(JMID),XCEND,YL(JMID),XP(IFST)
      WRITE(LU,102)IX(IMID),IX(ILST),IY(JMID),IY(JMID),
     &              XCEND,YL(JMID),XCEND+RAD1,YL(JMID),XP(IMID)
      WRITE(LU,102)IX(ILST),IX(IAFT),IY(JMID),IY(JMID),
     &              XCEND+RAD1,YL(JMID),XL(IAFT),YL(JMID),XP(ILST)
C
      CALL WRTFI(LU,IRX,IRY,IX,IY)
C
C-pd---Fix points around circle and certain ones inside--------------
C
      WRITE(LU,*)
      WRITE(LU,105)IX(1),IX(IRX+1),IY(1),IY(JFST)
      WRITE(LU,105)IX(1),IX(IFST),IY(JFST),IY(JLST)
      WRITE(LU,105)IX(ILST),IX(IRX+1),IY(JFST),IY(JLST)
      WRITE(LU,105)IX(1),IX(IRX+1),IY(JLST),IY(IRY+1)
C
      ISOL=4
      WRITE(LU,105)IX(IFST)+ISOL,IX(ILST)-ISOL,IY(JFST),IY(JLST)
      WRITE(LU,105)IX(IFST),IX(ILST),IY(JFST)+ISOL,IY(JLST)-ISOL
C
  564 CONTINUE
C
C*********************************************************************
C-pd---This section is for the constant cross sectional area of the---
C---   augmenter tube.  This cross section is located at the back-----
C---   side of the end wall.  Two options exist, one for a circle-----
C---   and one for a square.------------------------------------------
C    t5
      DO 565 I=1,2
C
      IX(1 )=1
      CALL SETIV(IX,IG,220,1,NI)
C
      XCENE=RG(47)
      YCENE=RG(48)
      RAD1=RG(59)/2.
      DXI=(RAD1*RAD1/2.)**0.5
      IFST=IG(237)
      JFST=IG(257)
      IMID=IFST+1
      JMID=JFST+1
      ILST=IFST+2
      JLST=JFST+2
      IBEF=IFST-1
      JBEF=JFST-1
      IAFT=IFST+3
      JAFT=JFST+3
C
      IRX=IG(48)
      IRY=IG(49)
C
      XL(1 )=0.0
      CALL SETRV(XL,RG,340,1,NI)
      XL(IFST  )=XCENE-DXI
      XL(IFST+2)=XCENE+DXI
C
      LU=68+IG(60)+I
      IF(XL(IRX+1).EQ.0.0) THEN
        XDEL=RG(LU+10)
```

```
         XL(IRX+1)=XCENE+(XCENE-XDEL)
      ENDIF

      CALL SETRV(XP,RG,360,2,NI)
C
      IY(1 )=1
      CALL SETIV(IY,IG,240,1 NI)
C
      YL(1 )=0.0
      CALL SETRV(YL,RG,380,1,NI)
      YL(JFST   )=YCENE-DXI
      YL(JFST+2)=YCENE+DXI
C
      CALL SETRV(YP,RG,400,2,NI)
C
      CG(LU)='CS  '
      I10=LU/10
      I1=LU-I10*10
      WRITE(CG(LU)(3:3),'(I1)') I10
      WRITE(CG(LU)(4:4),'(I1)') I1
      OPEN(LU,FILE=CG(LU),FORM='FORMATTED',STATUS='UNKNOWN')
      IF(RG(LU+10).NE.0.0) XL(1)=RG(LU+10)
      CALL WRTSQ(LU,NX,NY,IRX,IRY,IX,IY,XL,YL,XP,YP)
C
C-pd---Overwrite line info with arc data------------------------------
C
      ANG1=  0.0
      ANG2= 45.0
      ANG3= 90.0
      ANG4=135.0
      ANG5=180.0
      ANG6=225.0
      ANG7=270.0
      ANG8=315.0
      WRITE(LU,*)
      WRITE(LU,104)IX(IFST),IX(IMID),IY(JFST),IY(JFST),
     &             XCENE,YCENE,RAD1,ANG6,ANG7,XP(IFST)
      WRITE(LU,104)IX(IMID),IX(ILST),IY(JFST),IY(JFST),
     &             XCENE,YCENE,RAD1,ANG7,ANG8,XP(IMID)
      WRITE(LU,104)IX(IFST),IX(IMID),IY(JLST),IY(JLST),
     &             XCENE,YCENE,RAD1,ANG4,ANG3,XP(IFST)
      WRITE(LU,104)IX(IMID),IX(ILST),IY(JLST),IY(JLST),
     &             XCENE,YCENE,RAD1,ANG3,ANG2,XP(IMID)
      WRITE(LU,104)IX(IFST),IX(IFST),IY(JFST),IY(JMID),
     &             XCENE,YCENE,RAD1,ANG6,ANG5,YP(JFST)
      WRITE(LU,104)IX(IFST),IX(IFST),IY(JMID),IY(JLST),
     &             XCENE,YCENE,RAD1,ANG5,ANG4,YP(JMID)
      WRITE(LU,104)IX(ILST),IX(ILST),IY(JFST),IY(JMID),
     &             XCENE,YCENE,RAD1,ANG8,ANG1,YP(JFST)
      WRITE(LU,104)IX(ILST),IX(ILST),IY(JMID),IY(JLST),
     &             XCENE,YCENE,RAD1,ANG1,ANG2,YP(JMID)
C
C-pd---Shuffle lines------------------------------------------------
C
      WRITE(LU,102)IX(IMID),IX(IMID),IY(JBEF),IY(JFST),
     &             XL(IMID),YL(JBEF),XL(IMID),YCENE-RAD1,YP(JBEF)
      WRITE(LU,102)IX(IMID),IX(IMID),IY(JFST),IY(JMID),
     &             XL(IMID),YCENE-RAD1,XL(IMID),YCENE,YP(JFST)
      WRITE(LU,102)IX(IMID),IX(IMID),IY(JMID),IY(JLST),
     &             XL(IMID),YCENE,XL(IMID),YCENE+RAD1,YP(JMID)
```

```
      WRITE(LU,102)IX(IMID),IX(IMID),IY(JLST),IY(JAFT),
     &               XL(IMID),YCENE+RAD1,XL(IMID),YL(JAFT),YP(JLST)
      WRITE(LU,102)IX(IBEF),IX(IFST),IY(JMID),IY(JMID),
     &               XL(IBEF),YL(JMID),XCENE-RAD1,YL(JMID),XP(IBEF)
      WRITE(LU,102)IX(IFST),IX(IMID),IY(JMID),IY(JMID),
     &               XCENE-RAD1,YL(JMID),XCENE,YL(JMID),XP(IFST)
      WRITE(LU,102)IX(IMID),IX(ILST),IY(JMID),IY(JMID),
     &               XCENE,YL(JMID),XCENE+RAD1,YL(JMID),XP(IMID)
      WRITE(LU,102)IX(ILST),IX(IAFT),IY(JMID),IY(JMID),
     &               XCENE+RAD1,YL(JMID),XL(IAFT),YL(JMID),XP(ILST)
C
      CALL WRTFI(LU,IRX,IRY,IX,IY)
C
C-pd---Fix points around circle and certain ones inside---------------
C
      WRITE(LU,*)
      WRITE(LU,105)IX(1),IX(IRX+1),IY(1),IY(JFST)
      WRITE(LU,105)IX(1),IX(IFST),IY(JFST),IY(JLST)
      WRITE(LU,105)IX(ILST),IX(IRX+1),IY(JFST),IY(JLST)
      WRITE(LU,105)IX(1),IX(IRX+1),IY(JLST),IY(IRY+1)
C
      ISOL=4
      WRITE(LU,105)IX(IFST)+ISOL,IX(ILST)-ISOL,IY(JFST),IY(JLST)
      WRITE(LU,105)IX(IFST),IX(ILST),IY(JFST)+ISOL,IY(JLST)-ISOL
C
  565 CONTINUE
C
      J1TMP=IY(JFST)
      J2TMP=IY(JLST)
      YDTMP=YL(IRY)
      YDTOP=YL(IRY+1)
C
C
C*********************************************************************
C-pd---This section is for the exit of the chimney.  Uniform spacing--
C---   in each direction is assumed.---------------------------------
C    t6
      IX(1)=1
      IX(2)=NX+1
C
      XL(1)=XL(1)
      XL(2)=XL(IRX+1)
C
      XP(1)=1.0
C
      IY(1)=1
      IY(2)=NY+1
C
      YL(1)=RG(510+IG(537))
      YL(2)=RG(510+IG(537)-1)
C
      YP(1)=1.0
C
      LU=71+IG(60)
      CG(LU)='CS  '
      I10=LU/10
      I1=LU-I10*10
      WRITE(CG(LU)(3:3),'(I1)') I10
      WRITE(CG(LU)(4:4),'(I1)') I1
      OPEN(LU,FILE=CG(LU),FORM='FORMATTED',STATUS='UNKNOWN')
      IRX=1
      IRY=1
      CALL WRTSQ(LU,NX,NY,IRX,IRY,IX,IY,XL,YL,XP,YP)
      CALL WRTFI(LU,IRX,IRY,IX,IY)
C
      IF(IG(1).EQ.0) THEN
        WRITE(6,*)' TOTAL NUMBER OF GRID INPUT FILES CREATED = ',LU-60
        WRITE(6,*)' AT THIS POINT USE GGP TO CREATE GRID PLANES'
        RETURN
      ENDIF
C
C
C**********************************************************************
C**********************************************************************
C-pd---Call ggp--------------------------------------------------------
C
C         INACTIVE
C
C
C
C**********************************************************************
C**********************************************************************
C
      WRITE(6,*)' CREATING READCO FILE'
C
C-pd---Stack grids (NOTE:  SFAC hardwired in - SATLIT call before-----
C---    conversions set in Q1)----------------------------------------
C
      SFAC=0.0254
      NI=25
      LMX=(NX+1)*(NY+1)
      CALL SETIV(NZC,IG,510,3,NI)
      ZL(1 )=0.0
      CALL SETRV(ZL,RG,510,1,NI)
      CALL SETRV(ZP,RG,540,2,NI)
      CALL SETIV(IZT,IG,540,3,NI)
      CALL SETIV(IZF1,IG,570,3,NI)
      CALL SETIV(IZF2,IG,600,3,NI)
C
      LUW1=88
      OPEN(LUW1,FILE='grid',FORM='FORMATTED',STATUS='UNKNOWN')
      WRITE(LUW1,366)NX+1,NY+1,NZ+1
C
      DO 5005 I=1,IG(501)
      IF(IZT(I).EQ.1) THEN
        CALL XSTACK(CG(IZF1(I)),LMX,NZC(I),ZL(I),ZL(I+1),ZP(I),
     &              XAS1,YAS1,ZASL,SFAC,LUW1)
      ELSEIF (IZT(I).EQ.2) THEN
        CALL XBLEND(CG(IZF1(I)),CG(IZF2(I)),LMX,NZC(I),ZL(I),ZL(I+1),
     &              ZP(I),XAS,YAS,XAS1,YAS1,XAS2,YAS2,ZASL,SFAC,LUW1)
      ELSEIF (IZT(I).EQ.3) THEN
        ITRI=IG(90)
        ZPT=RG(90)
        NZC(I)=NZC(I)-ITRI
        CALL XCURVE(CG(IZF1(I)),LMX,NZC(I),ZL(I),ZL(I+1),ZP(I),YDTOP,
     &              J1TMP,J2TMP,ITRI,ZPT,YAS,ZAS,XAS1,YAS1,SFAC,LUW1)
      ELSEIF (IZT(I).EQ.4) THEN
        ZCH=RG(91)
        IF(ITOP.GT.ZCH) WRITE(6,*)'  ERROR:  EXIT OF CHIMNEY LOWER THAN
     &TOP OF ROOF  -->  CHECK DATA '
```

```
        CALL XLASTS(CG(IZF2(I)),LMX,NZC(I),ZCH,
     &              XAS,ZAS,XAS1,ZAS1,XAS2,ZAS2,YAS,YAS1,SFAC,LUW1)
      ELSE
        WRITE(6,*)' ERROR IN STACKING TYPE '
      ENDIF
 5005 CONTINUE
      CLOSE(LUW1,STATUS='KEEP')
C
      RETURN
C
  102 FORMAT('LI',4I3,F12.6,3F11.6,F7.2)
  103 FORMAT('FI',4I3)
  104 FORMAT('AR',4I3,F12.6,4F11.6,F7.2)
  105 FORMAT('FXY'4I3)
  366 FORMAT(3I5)
C
C--- GROUP 6. Body-fitted coordinates or grid distortion
    6 CONTINUE
      RETURN
C
C--- GROUP 7. Variables stored, solved & named
    7 CONTINUE
      RETURN
C
C--- GROUP 8. Terms (in differential equations) & devices
    8 CONTINUE
      RETURN
C
C--- GROUP 9. Properties of the medium (or media)
    9 CONTINUE
C
      IF(IG(1).NE.3) RETURN
      WRITE(6,*)' CALCULATING BOUNDARY CONDITIONS'
C
C-pd---Ambient---------------------------------------------------
      RGAS=RG(25)
      SC(1)=RG(1)/RG(21)
      SC(2)=RG(2)/RG(22)
      SC(3)=RG(3)/RG(23)
      SC(4)=RG(4)/RG(24)
      TEMP=RG(9)
      CALL ENTHAL(TEMP,HSUM,CPSUM,SC,4,0)
      RG(11)=CPSUM*RGAS*TEMP
C-pd---Engine----------------------------------------------------
      SC(1)=RG(5)/RG(21)
      SC(2)=RG(6)/RG(22)
      SC(3)=RG(7)/RG(23)
      SC(4)=RG(8)/RG(24)
      TEMP=RG(10)
      CALL ENTHAL(TEMP,HSUM,CPSUM,SC,4,0)
      RG(12)=CPSUM*RGAS*TEMP
      RETURN
C
C--- GROUP 10. Inter-phase-transfer processes and properties
   10 CONTINUE
      RETURN
C
C--- GROUP 11. Initialization of variable or porosity fields
   11 CONTINUE
      RETURN
```

```
C
C--- GROUP 12. Convection and diffusion adjustments
   12 CONTINUE
      RETURN
C
C--- GROUP 13. Boundary conditions and special sources
   13 CONTINUE
      RETURN
C
C--- GROUP 14. Downstream pressure for PARAB=.TRUE.
   14 CONTINUE
      RETURN
C
C--- GROUP 15. Termination of sweeps
   15 CONTINUE
      RETURN
C
C--- GROUP 16. Termination of iterations
   16 CONTINUE
      RETURN
C
C--- GROUP 17. Under-relaxation devices
   17 CONTINUE
      RETURN
C
C--- GROUP 18. Limits on variables or increments to them
   18 CONTINUE
      RETURN
C
C--- GROUP 19. Data communicated by satellite to GROUND
   19 CONTINUE
      RETURN
C
C--- GROUP 20. Preliminary print-out
   20 CONTINUE
      RETURN
C
C--- GROUP 21. Print-out of variables
   21 CONTINUE
      RETURN
C
C--- GROUP 22. Spot-value print-out
   22 CONTINUE
      RETURN
C
C--- GROUP 23. Field print-out and plot control
   23 CONTINUE
      RETURN
C
C--- GROUP 24. Dumps for restarts
   24 CONTINUE
      WRITE(6,*)'                      OUT OF IT '
      RETURN
      END
C***********************************************************************
      SUBROUTINE GCALE(GFACT)
C***********************************************************************
C  GCALE gets information needed to scale grid points.
C---------------------------------------------------------------------
C
```

```
      INCLUDE 'satear'
      INCLUDE 'satloc'
C     INCLUDE 'bfcsat'
      COMMON F(1)
C
      NI=NX+1
      NJ=NY+1
      NK=NZ+1
      JNNN=NI*NJ*NK
      CALL SCALEW(F(KXC+1),F(KYC+1),F(KZC+1),GFACT,JNNN)
C
      RETURN
      END
C
C**********************************************************************
      SUBROUTINE SCALEW(X,Y,Z,F,N)
C**********************************************************************
C  GCALEW converts grid nodes to the proper units (m).
C--------------------------------------------------------------------
C
      DIMENSION X(*),Y(*),Z(*)
C
      DO 1 I=1,N
      X(I)=X(I)*F
      Y(I)=Y(I)*F
    1 Z(I)=Z(I)*F
C
      RETURN
      END
C
C**********************************************************************
      SUBROUTINE ENTHAL(TEMP,HSUM,CPSUM,SC,NS,NFO)
C**********************************************************************
C  ENTHAL calculates H/RT from JANNAF data.  The order of
C  species is N O C H.
C--------------------------------------------------------------------
C
      DIMENSION SC(4),ZS(7,2,4)
      DATA ZS/ 0.28532899E+01,  0.16022128E-02, -0.62936893E-06,
     &         0.11441022E-09, -0.78057465E-14, -0.89008093E+03,
     &         0.63964897E+01,  0.37044177E+01, -0.14218753E-02,
     &         0.28670392E-05, -0.12028885E-08, -0.13954677E-13,
     &        -0.10640795E+04,  0.22336285E+01,
     &         0.36122139E+01,  0.74853166E-03, -0.19820647E-06,
     &         0.33749008E-10, -0.23907374E-14, -0.11978151E+04,
     &         0.36703307E+01,  0.37837135E+01, -0.30233634E-02,
     &         0.99492751E-05, -0.98189101E-08,  0.33031825E-11,
     &        -0.10638107E+04,  0.36416345E+01,
     &         0.44608041E+01,  0.30981719E-02, -0.12392571E-05,
     &         0.22741325E-09, -0.15525954E-13, -0.48961442E+05,
     &        -0.98635982E+00,  0.24007797E+01,  0.87350957E-02,
     &        -0.66070878E-05,  0.20021861E-08,  0.63274039E-15,
     &        -0.48377527E+05,  0.96951457E+01,
     &         0.27167633E+01,  0.29451374E-02, -0.80224374E-06,
     &         0.10226682E-09, -0.48472145E-14, -0.29905826E 05,
     &         0.66305671E+01,  0.40701275E+01, -0.11084499E-02,
     &         0.41521180E-05, -0.29637404E-08,  0.80702103E-12,
     &        -0.30279722E+05, -0.32270046E+00 /
C
      K=1
      IF(TEMP.LT.1000.) K=2
      TEMP2=TEMP*TEMP
      HSUM=0.
      CPSUM=0.
      DO 100 IS=1,NS
      CP1=ZS(1,K,IS)
      CP2=ZS(2,K,IS)*TEMP
      CP3=ZS(3,K,IS)*TEMP2
      CP4=ZS(4,K,IS)*TEMP2*TEMP
      CP5=ZS(5,K,IS)*TEMP2*TEMP2
      CPSUM=CPSUM+SC(IS)*(CP1+CP2+CP3+CP4+CP5)
  100 HSUM =HSUM+
     1  SC(IS)*(CP1+.5*CP2+.33333*CP3+.25*CP4+.2*CP5+ZS(6,K,IS)/TEMP)
C
      RETURN
      END
C
C***********************************************************************
      SUBROUTINE SETIV(IA,IG,IFST,ITY,NI)
C***********************************************************************
C SETIV places integer values from the IG array into the
C proper local array.
C---------------------------------------------------------------------
C
      DIMENSION IA(*),IG(*)
C
      IF(ITY.EQ.1) THEN
        DO 1 I=1,NI
    1   IA(I+1)=IG(IFST+I)+1
      ELSEIF (ITY.EQ.2) THEN
        DO 2 I=1,NI
    2   IA(I)=IG(IFST+I)+1
      ELSEIF (ITY.EQ.3) THEN
        DO 3 I=1,NI
    3   IA(I)=IG(IFST+I)
      ELSE
        WRITE(6,*)' ERROR SETIV --- INVALID TYPE '
      ENDIF
C
      RETURN
      END
C
C***********************************************************************
      SUBROUTINE SETRV(RA,RG,IFST,ITY,NI)
C***********************************************************************
C SETRV places real values from the RG array into the proper
C local array.
C---------------------------------------------------------------------
C
      DIMENSION RA(*),RG(*)

      IF(ITY.EQ.1) THEN
        DO 1 I=1,NI
    1   RA(I+1)=RG(IFST+I)
      ELSEIF (ITY.EQ.2) THEN
        DO 2 I=1,NI
    2   RA(I)=RG(IFST+I)
      ELSE
        WRITE(6,*)' ERROR SETRV --- INVALID TYPE '
      ENDIF
```

```
C
      RETURN
      END
C
C************************************************************************
      SUBROUTINE WRTSQ(LU,NX,NY,IRX,IRY,IX,IY,XL,YL,XP,YP)
C************************************************************************
C  WRTSQ writes input grid file assuming all straight lines.
C----------------------------------------------------------------------
C
      DIMENSION IX(*),IY(*),XL(*),YL(*),XP(*),YP(*)
C
      WRITE(LU,100) NX+1
      WRITE(LU,101) NY+1
      DO 10 I=1,IRY+1
      WRITE(LU,*)
      DO 10 J=1,IRX
   10 WRITE(LU,102)
     &     IX(J),IX(J+1),IY(I),IY(I),XL(J),YL(I),XL(J+1),YL(I),XP(J)
      DO 20 I=1,IRX+1
      WRITE(LU,*)
      DO 20 J=1,IRY
   20 WRITE(LU,102)
     &     IX(I),IX(I),IY(J),IY(J+1),XL(I),YL(J),XL(I),YL(J+1),YP(J)
C
  100 FORMAT('IMAX',I3)
  101 FORMAT('JMAX',I3)
  102 FORMAT('LI',4I3,F12.6,3F11.6,F7.2)
C
      RETURN
      END
C
C************************************************************************
      SUBROUTINE WRTFI(LU,IRX,IRY,IX,IY)
C************************************************************************
C  WRTFI writes commands needed to fill subsections.
C----------------------------------------------------------------------
C
      DIMENSION IX(*),IY(*)
C
      DO 10 I=1,IRY
      WRITE(LU,*)
      DO 10 J=1,IRX
   10 WRITE(LU,103)IX(J),IX(J+1),IY(I),IY(I+1)
C
  103 FORMAT('FI',4I3)
C
      RETURN
      END
C
C************************************************************************
      SUBROUTINE WRTFI2(LU,IRX,IRY,IX,IY,IF,JF)
C************************************************************************
C  WRTFI writes commands needed to fill subsections.
C----------------------------------------------------------------------
C
      DIMENSION IX(*),IY(*)
C
      DO 10 I=1,IRY
      WRITE(LU,*)
      DO 10 J=1,IRX
      IF((I.EQ.JF.OR.I.EQ.JF+1).AND.(J.EQ.IF.OR.J.EQ.IF+1)) GOTO 10
      WRITE(LU,103)IX(J),IX(J+1),IY(I),IY(I+1)
   10 CONTINUE
C
  103 FORMAT('FI',4I3)
C
      RETURN
      END
C
C*****************************************************************
      SUBROUTINE XSTACK(F1PRE,LMX,NZC,ZFST,ZLST,ZP,X1,Y1,ZL,CV,LUW1)
C*****************************************************************
C XSTACK repeats one computational grid file
C----------------------------------------------------------------
C
      CHARACTER*4 F1PRE,FEXT
      CHARACTER*8 F1NAME
      DIMENSION X1(*),Y1(*),ZL(*)
C
      FEXT='.GRD'
      F1NAME=F1PRE//FEXT
      LUR1=80
      OPEN(LUR1,FILE=F1NAME,FORM='FORMATTED',STATUS='OLD')
C
      READ(LUR1,366)LP1,MP1,NTP1
      READ(LUR1,333)((X1(IJ),IJ=I,LMX,LP1),I=1,LP1)
      READ(LUR1,333)((Y1(IJ),IJ=I,LMX,LP1),I=1,LP1)
      READ(LUR1,333)((ZTEMP, IJ=I,LMX,LP1),I=1,LP1)
C
      CALL ZLSET(ZL,1,NZC+1,ZFST,ZLST,ZP)
C
      DO 10 K=1,NZC
      WRITE(LUW1,333)((X1(IJ)*CV,IJ=I,LMX,LP1),I=1,LP1)
      WRITE(LUW1,333)((Y1(IJ)*CV,IJ=I,LMX,LP1),I=1,LP1)
      WRITE(LUW1,333)((ZL(K)*CV, IJ=I,LMX,LP1),I=1,LP1)
   10 CONTINUE
      CLOSE(LUR1,STATUS='KEEP')
C
      RETURN
  333 FORMAT(5(1P,E13.6))
  366 FORMAT(3I5)
      END
C
C*****************************************************************
      SUBROUTINE XBLEND(F1PRE,F2PRE,LMX,NZC,ZFST,ZLST,ZP,X,Y,X1,Y1,
     &                  X2,Y2,ZL,CV,LUW1)
C*****************************************************************
C XBLEND blends two computational grids files
C----------------------------------------------------------------
C
      CHARACTER*4 F1PRE,F2PRE,FEXT
      CHARACTER*8 F1NAME,F2NAME
      DIMENSION X(2500),Y(2500),X1(2500),Y1(2500),X2(2500),Y2(2500),
     &          ZL(100)
C
      FEXT='.GRD'
      F1NAME=F1PRE//FEXT
      F2NAME=F2PRE//FEXT
      LUR1=80
      LUR2=81
      OPEN(LUR1,FILE=F1NAME,FORM='FORMATTED',STATUS='OLD')
      OPEN(LUR2,FILE=F2NAME,FORM='FORMATTED',STATUS='OLD')
C
      READ(LUR1,366)LP1,MP1,NTP1
      READ(LUR1,333)((X1(IJ),IJ=I,LMX,LP1),I=1,LP1)
      READ(LUR1,333)((Y1(IJ),IJ=I,LMX,LP1),I=1,LP1)
      READ(LUR1,333)((ZTEMP, IJ=I,LMX,LP1),I=1,LP1)
      READ(LUR2,366)LP1,MP1,NTP1
      READ(LUR2,333)((X2(IJ),IJ=I,LMX,LP1),I=1,LP1)
      READ(LUR2,333)((Y2(IJ),IJ=I,LMX,LP1),I=1,LP1)
      READ(LUR2,333)((ZTEMP, IJ=I,LMX,LP1),I=1,LP1)
C
      CALL ZLSET(ZL,1,NZC+1,ZFST,ZLST,ZP)
C
      DO 20 K=1,NZC
      DO 21 I=1,LMX
      IF(NZC.EQ.1) THEN
        X(I)=X1(I)
        Y(I)=Y1(I)
      ELSE
        X(I)=X1(I)*FLOAT(NZC-K+1)/FLOAT(NZC)+
     &        X2(I)*FLOAT(K-1)/FLOAT(NZC)
        Y(I)=Y1(I)*FLOAT(NZC-K+1)/FLOAT(NZC)+
     &        Y2(I)*FLOAT(K-1)/FLOAT(NZC)
      ENDIF
   21 CONTINUE
      WRITE(LUW1,333)((X(IJ)*CV,IJ=I,LMX,LP1),I=1,LP1)
      WRITE(LUW1,333)((Y(IJ)*CV,IJ=I,LMX,LP1),I=1,LP1)
      WRITE(LUW1,333)((ZL(K)*CV,IJ=I,LMX,LP1),I=1,LP1)
   20 CONTINUE
      CLOSE(LUR1,STATUS='KEEP')
      CLOSE(LUR2,STATUS='KEEP')
C
      RETURN
  333 FORMAT(5(1P,E13.6))
  366 FORMAT(3I5)
      END
C
C***********************************************************************
      SUBROUTINE XCURVE(F1PRE,LMX,NZC,ZFST,ZLST,ZP,CENC,NY2,NY3,ITRI,
     &                  ZPT,Y,Z,X1,Y1,CV,LUW1)
C***********************************************************************
C  XCURVE creates the grid in the augmenter tube bend section
C---------------------------------------------------------------------
C
      CHARACTER*4 F1PRE,FEXT
      CHARACTER*8 F1NAME
      DIMENSION Y(*),Z(*),X1(*),Y1(*)
C
      FEXT='.GRD'
      F1NAME=F1PRE//FEXT
      LUR1=80
C-pd---NZC number of cells in bend  (WARNING: Must be even)-----------
C---   NY1 lower Y line-----------------------------------------------
C---   NY2 lower Y circle line----------------------------------------
C---   NY3 upper Y circle line----------------------------------------
C---   NY5 upper Y line-----------------------------------------------
C
      OPEN(LUR1,FILE=F1NAME,FORM='FORMATTED',STATUS='OLD')
      READ(LUR1,366)LP1,MP1,NTP1
      READ(LUR1,333)((X1(IJ),IJ=I,LMX,LP1),I=1,LP1)
      READ(LUR1,333)((Y1(IJ),IJ=I,LMX,LP1),I=1,LP1)
      READ(LUR1,333)((ZTEMP ,IJ=I,LMX,LP1),I=1,LP1)
C
      NY1=1
      NY5=MP1
C
C-pd---Do Boundary------------------------------------------------------
      WRITE(LUW1,333)((X1(IJ)*CV,IJ=I,LMX,LP1),I=1,LP1)
      WRITE(LUW1,333)((Y1(IJ)*CV,IJ=I,LMX,LP1),I=1,LP1)
      WRITE(LUW1,333)((ZFST*CV,  IJ=I,LMX,LP1),I=1,LP1)
C
C-pd---Do Straight section of pipe--------------------------------------
      DO 100 IP=1,ITRI
      DELZ=ZPT*FLOAT(IP)/ITRI
C
      DO 105 J=1,MP1-1
      DO 105 I=1,LP1
      LOC=(J-1)*LP1+I
  105 Y(LOC)=Y1(LOC)
C
      DO 110 I=1,LP1
      LOC=(MP1-1)*LP1+I
  110 Y(LOC)=Y1(LOC)+0.001*FLOAT(IP)
C
      DO 115 J=1,NY3
      DO 115 I=1,LP1
      LOC=(J-1)*LP1+I
  115 Z(LOC)=ZFST+DELZ
C
      YFST=Y1(NY3*LP1)
      YLST=Y1(MP1*LP1)
      DO 120 J=NY3+1,MP1
      DO 120 I=1,LP1
      LOC=(J-1)*LP1+I
      YLOC=Y1(J*LP1)
      YFCT=1.0-(YLOC-YFST)/(YLST-YFST)
  120 Z(LOC)=ZFST+DELZ*YFCT
C
      WRITE(LUW1,333)((X1(IJ)*CV,IJ=I,LMX,LP1),I=1,LP1)
      WRITE(LUW1,333)((Y(IJ)*CV, IJ=I,LMX,LP1),I=1,LP1)
      WRITE(LUW1,333)((Z(IJ)*CV, IJ=I,LMX,LP1),I=1,LP1)
  100 CONTINUE
C
C-pd---Do curve section-------------------------------------------------
      ZFST=ZFST+ZPT
      ZLEN=ZLST-ZFST
      DO 400 IP=1,NZC
      ANG=90.0/FLOAT(NZC)*FLOAT(IP)
      PI=3.141592654
      RAD=ANG/360.*2.*PI
      YFAC=COS(RAD)

C-pd---Lower Y row------------------------------------------------------
      DO 205 I=1,LP1
      IF(IP.LE.NZC/2) THEN
        Y(I)=0.0
      ELSE
        Y(I)=FLOAT(IP-(NZC/2))/FLOAT(NZC/2)*CENC
      ENDIF
  205 CONTINUE
C
C-pd---Lower Y circle row-------------------------------------------
      IAD=(NY2-1)*LP1
      DO 210 I=1,LP1
      Y(IAD+I)=Y1(IAD+I)+(1.0-YFAC)*(CENC-Y1(IAD+I))
  210 CONTINUE
C
C-pd---Upper Y circle row-------------------------------------------
      IAD=(NY3-1)*LP1
      DO 215 I=1,LP1
      Y(IAD+I)=Y1(IAD+I)+(1.0-YFAC)*(CENC-Y1(IAD+I))
  215 CONTINUE
C
C-pd---Upper Y row--------------------------------------------------
      IAD=(MP1-1)*LP1
      XFUG=((FLOAT(IP)/FLOAT(NZC))*0.01)+(0.001*FLOAT(ITRI))
      DO 220 I=1,LP1
C-pd---add fact to give a north cell area---------------------------
      Y(IAD+I)=Y1(IAD+I)+XFUG
  220 CONTINUE
C
C-pd---Fill first section-------------------------------------------
      DO 250 J=2,NY2-1
      DO 250 I=1,LP1
      LOC=(J-1)*LP1+I
      IAD1=0
      IAD2=(NY2-1)*LP1
      Y(LOC)=Y(IAD1+I)+((Y1(LOC)-Y1(IAD1+I))/
     +        (Y1(IAD2+I)-Y1(IAD1+I))*(Y(IAD2+I)-Y(IAD1+I)))
  250 CONTINUE
C
C-pd---Fill circle section------------------------------------------
      DO 260 J=NY2+1,NY3-1
      DO 260 I=1,LP1
      LOC=(J-1)*LP1+I
      IAD1=(NY2-1)*LP1
      IAD2=(NY3-1)*LP1
      Y(LOC)=Y(IAD1+I)+((Y1(LOC)-Y1(IAD1+I))/
     +        (Y1(IAD2+I)-Y1(IAD1+I))*(Y(IAD2+I)-Y(IAD1+I)))
  260 CONTINUE
C
C-pd---Fill top section---------------------------------------------
      DO 270 J=NY3+1,MP1-1
      DO 270 I=1,LP1
      LOC=(J-1)*LP1+I
      IAD1=(NY3-1)*LP1
      IAD2=(MP1-1)*LP1
      Y(LOC)=Y(IAD1+I)+((Y1(LOC)-Y1(IAD1+I))/
     +        (Y1(IAD2+I)-Y1(IAD1+I))*(Y(IAD2+I)-Y(IAD1+I)))
  270 CONTINUE
C
C
      ZD4=0.0
C
C-pd---Lower Z row--------------------------------------------------
      ZFAC=SIN(RAD)
      DO 305 I=1,LP1
      IF(IP.LE.NZC/2) THEN
```

```
        Z(I)=FLOAT(IP)/FLOAT(NZC/2)*ZLEN+ZFST
      ELSE
        Z(I)=ZLEN+ZFST
      ENDIF
  305 CONTINUE
C
C-pd---Lower Z circle row-----------------------------------------
      IAD=(NY2-1)*LP1
      DO 310 I=1,LP1
      Z(IAD+I)=SIN(RAD)*(CENC-Y1(IAD+I))+ZFST
  310 CONTINUE
C
C-pd---Upper Z circle row-----------------------------------------
      IAD=(NY3-1)*LP1
      DO 315 I=1,LP1
      Z(IAD+I)=SIN(RAD)*(CENC-Y1(IAD+I))+ZFST
  315 CONTINUE
C
C-pd---Upper Z row------------------------------------------------
      IAD=(MP1-1)*LP1
      DO 320 I=1,LP1
      Z(IAD+I)=ZD4+ZFST-ZPT
  320 CONTINUE
C
C-pd---Fill first section-----------------------------------------
      DO 350 J=2,NY2-1
      DO 350 I=1,LP1
      LOC=(J-1)*LP1+I
      IAD1=0
      IAD2=(NY2-1)*LP1
      Z(LOC)=Z(IAD1+I)-((Y1(LOC)-Y1(IAD1+I))/
     +        (Y1(IAD2+I)-Y1(IAD1+I))*(Z(IAD1+I)-Z(IAD2+I)))
  350 CONTINUE
C
C-pd---Fill circle section----------------------------------------
      DO 360 J=NY2+1,NY3-1
      DO 360 I=1,LP1
      LOC=(J-1)*LP1+I
      IAD1=(NY2-1)*LP1
      IAD2=(NY3-1)*LP1
      Z(LOC)=Z(IAD1+I)-((Y1(LOC)-Y1(IAD1+I))/
     +        (Y1(IAD2+I)-Y1(IAD1+I))*(Z(IAD1+I)-Z(IAD2+I)))
  360 CONTINUE
C
C-pd---Fill top section-------------------------------------------
      DO 370 J=NY3+1,MP1-1
      DO 370 I=1,LP1
      LOC=(J-1)*LP1+I
      IAD1=(NY3-1)*LP1
      IAD2=(MP1-1)*LP1
      Z(LOC)=Z(IAD1+I)-((Y1(LOC)-Y1(IAD1+I))/
     +        (Y1(IAD2+I)-Y1(IAD1+I))*(Z(IAD1+I)-Z(IAD2+I)))
  370 CONTINUE

C-pd---Write data-------------------------------------------------
      WRITE(LUW1,333)((X1(IJ)*CV,IJ=I,LMX,LP1),I=1,LP1)
      WRITE(LUW1,333)((Y(IJ)*CV, IJ=I,LMX,LP1),I=1,LP1)
      WRITE(LUW1,333)((Z(IJ)*CV, IJ=I,LMX,LP1),I=1,LP1)
  400 CONTINUE
```

```
      CLOSE(LUR1,STATUS='KEEP')
C
      RETURN
C
  333 FORMAT(5(1P,E13.6))
  366 FORMAT(3I5)
C
      END
C
C*********************************************************************
      SUBROUTINE XLASTS(F1PRE,LMX,NZC,YC,
     &                  X,Z1,X1,Z,X2,Z2,Y1,Y,CV,LUW1)
C*********************************************************************
C  XLASTS creates the grid in the last section
C-------------------------------------------------------------------
C
      CHARACTER*4 F1PRE,F2PRE,FEXT
      CHARACTER*8 F1NAME
      DIMENSION X(*),Z(*),X1(*),Z1(*),X2(*),Z2(*),Y1(*),Y(*)
C
      FEXT='.GRD'
      F1NAME=F1PRE//FEXT
      LUR1=80
C
C-pd---Do last section (blend)-----------------------------------
      OPEN(LUR1,FILE=F1NAME,FORM='FORMATTED',STATUS='OLD')
      READ(LUR1,366)LP1,MP1,NTP1
      READ(LUR1,333)((X2(IJ),IJ=I,LMX,LP1),I=1,LP1)
      READ(LUR1,333)((Z2(IJ),IJ=I,LMX,LP1),I=1,LP1)
      READ(LUR1,333)((ZTEMP ,IJ=I,LMX,LP1),I=1,LP1)
      DO 440 K=2,NZC+1
      DO 441 I=1,LMX
      IF(NZC.EQ.1) THEN
        X(I)=X1(I)
        Z(I)=Z1(I)
      ELSE
        X(I)=X1(I)*FLOAT(NZC-K+1)/FLOAT(NZC)+
     &          X2(I)*FLOAT(K-1)/FLOAT(NZC)
        Y(I)=Y1(I)*FLOAT(NZC-K+1)/FLOAT(NZC)+
     &          YC*FLOAT(K-1)/FLOAT(NZC)
        Z(I)=Z1(I)*FLOAT(NZC-K+1)/FLOAT(NZC)+
     &          Z2(I)*FLOAT(K-1)/FLOAT(NZC)
      ENDIF
  441 CONTINUE
      WRITE(LUW1,333)((X(IJ)*CV,IJ=I,LMX,LP1),I=1,LP1)
      WRITE(LUW1,333)((Y(IJ)*CV,IJ=I,LMX,LP1),I=1,LP1)
      WRITE(LUW1,333)((Z(IJ)*CV,IJ=I,LMX,LP1),I=1,LP1)
  440 CONTINUE
      CLOSE(LUR1,STATUS='KEEP')
C
      RETURN
  333 FORMAT(5(1P,E13.6))
  366 FORMAT(3I5)
      END
C
C*********************************************************************
      SUBROUTINE ZLSET(ZBND,INDEX1,INDEXL,Z1,ZL,PWR)
C*********************************************************************
C
C (C) COPYRIGHT 1991 DOC D of North America, Inc.  ALL RIGHTS RESERVED
```

```
C
C  Read input parameters to distribute a number of points along a
   line segment.
C  Syntax is : LINE  K1  KL  Z1  ZL  ÄPWRÅ
C---------------------------------------------------------------------
      DIMENSION ZBND(*)

      IF(PWR.GT.0) THEN
        K1=INDEX1
        KL=INDEXL
        INC=1
        DELZ = ZL-Z1
        ZF = Z1
      ELSE
        K1=INDEXL
        KL=INDEX1
        INC=-1
        DELZ = Z1-ZL
        ZF = ZL
        PWR=ABS(PWR)
      ENDIF
      DO 10 I = K1,KL,INC
      RAT = (FLOAT(I-K1)/FLOAT(KL-K1))**PWR
      ZBND(I) = ZF + DELZ*RAT
   10 CONTINUE
C
      RETURN
      END
```

APPENDIX D

```
C FILE NAME GROUND.FTN--------------------------------22 April 87
C THIS IS THE MAIN PROGRAM OF EARTH
C
C  (C) COPYRIGHT 1984, LAST REVISION 1987.
C  CONCENTRATION HEAT AND MOMENTUM LTD. ALL RIGHTS RESERVED.
C  This subroutine and the remainder of the PHOENICS code are
C  proprietary software owned by Concentration Heat and Momentum
C  Limited, 40 High Street, Wimbledon, London SW19 5AU, England.
C
C
C     PROGRAM MAIN
C
C 1   The following two COMMON's, which appear identically in the
C     satellite MAIN program, allow up to 80 dependent variables to
C     be solved for (or their storage spaces to be occupied by
C     other variables, such as density). If a larger number is
C     required increase the parameter nvd. Less than 50 for nvd is not
C     permitted.
C
C     If more patches are required increase npatd.
C
C     If a larger F-array is needed increase nfd.
C
      PARAMETER (NVD=80,NFD=18000000,NPATD=1000)
C
      COMMON/LGE4/L4(NVD)
     1/LDB1/L5(NVD)/IDA1/I1(NVD)/IDA2/I2(NVD)/IDA3/I3(NVD)/IDA4/I4(NVD)
     1/IDA5/I5(NVD)/IDA6/I6(NVD)/GI1/I7(NVD)/GI2/I8(NVD)/HDA1/IH1(NVD)
     1/GH1/IH2(NVD)/RDA1/R1(NVD)/RDA2/R2(NVD)/RDA3/R3(NVD)/RDA4/R4(NVD)
     1/RDA5/R5(NVD)/RDA6/R6(NVD)/RDA7/R7(NVD)/RDA8/R8(NVD)/RDA9/R9(NVD)
     1/RDA10/R10(NVD)/RDA11/R11(NVD)
     1/GR1/R12(NVD)/GR2/R13(NVD)/GR3/R14(NVD)/GR4/R15(NVD)
     1/IPIP1/IP1(NVD)/HPIP2/IHP2(NVD)/RPIP1/RVAL(NVD)/LPIP1/LVAL(NVD)
     1/IFPL/IPL0(NVD)/RFPL1/ORPRIN(NVD)/RFPL2/ORMAX(NVD)
     1/RFPL3/ORMIN(NVD)
      LOGICAL L1,L2,L3,L4,L5,DBGFIL,LVAL
      CHARACTER*4 IH1,IH2,IHP2,NSDA
C
      COMMON/F01/I9(4*NVD)
      COMMON/DISC/DBGFIL
      COMMON/LUNITS/LUNIT(60)
      EXTERNAL WAYOUT
C
C 2   Set dimensions of data-for-GROUND arrays here. WARNING: the
C     corresponding arrays in the MAIN program of the satellite
C     (see SATLIT) must have the same dimensions.
      COMMON/LGRND/LG(1000)/IGRND/IG(1000)/RGRND/RG(10000)
      COMMON/CGRND/CG(1000)
      LOGICAL LG
      CHARACTER*4 CG
C
C 3   Set dimensions of data-for-GREX2 arrays here. WARNING: the
C     corresponding arrays in the MAIN program of the satellite
C     (see SATLIT) must have the same dimensions.
      COMMON/LSG/LSGD(20)/ISG/ISGD(20)/RSG/RSGD(100)/CSG/CSGD(10)
      LOGICAL LSGD
      CHARACTER*4 CSGD
C
C 4   Set dimension of patch-name array here. WARNING: the array
C     NAMPAT in the MAIN program of the satellite must have the
```

```
C     dimension.
      COMMON/NPAT/NAMPAT(NPATD)
      CHARACTER*8 NAMPAT
C
C     CONFIG FILE name declaration.
      COMMON/CNFG/CNFIG
      CHARACTER CNFIG*48
C
C 5   The numbers in the next two statements (which must be ident-
C     ical) indicate how much computer memory is to be set aside
C     for storing the main and auxiliary variables. The user may
C     alter them if he wishes, to accord with the number of
C     grid nodes and dependent variables he is concerned with.
      COMMON F(NFD)
      NFDIM=NFD
C
C 6   Logical-unit numbers and file names, not to be changed.
      CALL CNFGZZ(2)
      CALL EARSET(1)
      CALL OPENFL(6)
C
C     User may here change message transmitted to logical unit
C     LUPR3
      CALL WRIT40('Ground-Station is ground.f, 09/25/87.    ')
      CALL MAIN1(NFDIM)
      CALL WAYOUT(0)
      STOP
      END
C*****************************************************************
      SUBROUTINE GROSTA
C
      INCLUDE 'satear'
      INCLUDE 'grdloc'
      INCLUDE 'grdear'
C.... This subroutine directs control to the GROUNDs selected by
C     the satellite settings of USEGRX, NAMGRD & USEGRD.
C
C           Subroutine GREX2 contains options for fluid properties,
C     turbulence models, wall functions, chemical reaction etc. It
C     was introduced in version 1.4 of PHOENICS.
C
          IF(USEGRX) CALL GREX2
C
C.... BTSTGR contains the sequences used in conjunction with
C     the BFC test battery.
C
          IF(NAMGRD.EQ.'BTST') CALL BTSTGR
C
C.... TESTGR contains test battery sequences used in conjunction
C     with the test-battery SATLIT subroutine, TESTST.
C
          IF(NAMGRD.EQ.'TEST') CALL TESTGR
C
C.... SPECGR is a generic "special" GROUND the name of which can
C     be used by anyone for their own purposes. SPC1GR, SPC2GR and
C     SPC3GR permit the user to attach his own library of special
C     GROUNDs selected according to the prescription of NAMGRD.
C
          IF(NAMGRD.EQ.'SPEC') CALL SPECGR
C
```

```
C.... The subroutine GROUND attached to the bottom of this file is
C     an unallocated blank form into which the user can insert his
C     own FORTRAN sequences. The PIL parameter USEGRD governs entry
C     in to it.
C
          IF(USEGRD) CALL GROUND
C
C.... The data echo is called at the preliminary print-out stage.
          IF(IGR.NE.20) RETURN
          IF(.NOT.ECHO) GO TO 20
      CALL DATPRN(Y,Y,Y,Y,  Y,Y,Y,Y,  Y,Y,Y,N,  Y,Y,Y,Y,
     &            Y,Y,Y,Y,  Y,Y,Y,Y)
      RETURN
   20 CALL DATPRN(Y,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N)
      RETURN
      END
      SUBROUTINE SPECGR
      CALL WRIT40('DUMMY SUBROUTINE SPECGR CALLED.        ')
      CALL WRIT40('PLEASE ATTACH SPECGR OBJECT AT LINK.   ')
      CALL WAYOUT(2)
      RETURN
      END
      SUBROUTINE QUIZ
      RETURN
      END
C****************************************************************
      SUBROUTINE GROUND
C
      INCLUDE 'satear'
      INCLUDE 'grdloc'
      INCLUDE 'grdear'
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION STARTS:
C
C 1   Set dimensions of data-for-GROUND arrays here. WARNING: the
C     corresponding arrays in the MAIN program of the satellite
C     and EARTH must have the same dimensions.
      COMMON/LGRND/LG(1000)/IGRND/IG(1000)/RGRND/RG(10000)
      COMMON/CGRND/CG(1000)
      COMMON/GR3/RESD(1)
      LOGICAL LG,DONE
      CHARACTER*4 CG,ADIR*1,ANUX*1
      DATA DONE /.FALSE./
      INTEGER TEMP,CP,PH2O,TFAR,RHOE,SPAR
C
C 2   User dimensions own arrays here, for example:
C     DIMENSION UUH(10,10),UUC(10,10),UUX(10,10),UUZ(10)
      PARAMETER (JNX=45,JNY=40,JNXY=JNX*JNY)
      PARAMETER (NDATA=15,NCURVES=5)
      DIMENSION GAH(JNY,JNX),GP1(JNY,JNX),GH1(JNY,JNX),GC1(JNY,JNX),
     &          GC2(JNY,JNX),GC3(JNY,JNX),GRH(JNY,JNX),GTMP(JNY,JNX),
     &          GVPR(JNY,JNX),GCP(JNY,JNX),PHI(JNY,JNX),A1(JNXY),
     &          A2(JNXY),A3(JNXY),A4(JNXY),A5(JNXY),A6(JNXY),
     &          EFX(JNXY),EFY(JNXY),FMAG(JNXY),RP(JNXY),
     &          CTDATA(NDATA,NCURVES)
      DIMENSION SC(4)
C
C 3   User places his data statements here, for example:
C     DATA NXDIM,NYDIM/10,10/
      DATA CTDATA /0.10,0.15,0.20,0.25 ,0.30,0.35,0.40,0.45,0.50 ,0.55,
     &             0.60,0.65,0.70,0.75 ,0.0,
     &             0.00,0.00,1.60,1.375,1.22,1.08,0.96,0.86,0.775,0.70,
     &                           5*0.00,
     &             0.00,1.68,1.42,1.22 ,1.08,0.95,0.86,0.775,0.71,0.64,
     &             0.58,0.52,0.46,0.43 ,0.00,
     &                          15*0.00,
     &                          15*0.00/
      EQUIVALENCE (TEMP,C4),(CP,C5),(PH2O,C8),(TFAR,C9),(RHOE,C10),
     &            (SPAR,C11)
C
C 4   Insert own coding below as desired, guided by GREX2 examples.
C     Note that the satellite-to-GREX2 special data in the labelled
C     COMMONs /RSG/, /ISG/, /LSG/ and /CSG/ (which are now automatically
C     included in grdloc) can be used but the user must check GREX2 for
C     any conflicting uses. The same comment applies to the EARTH-spare
C     working arrays EASP1, EASP2,....EASP10. If the call to GREX2 has
C     been deactivated then they can all be used without reservation.
C
      IXL=IABS(IXL)
          IF(IGR.EQ.13) GO TO 13
          IF(IGR.EQ.19) GO TO 19
      GO TO (1,2,3,4,5,6,24,8,9,10,11,12,13,14,24,24,24,24,19,20,24,
     124,23,24),IGR
C*****************************************************************
C
C--- GROUP 1. Run title and other preliminaries
C
    1 GO TO (1001,1002),ISC
 1001 CONTINUE
C
      NSC=4
      NFO=0
      TNY=1.E-15
      RGAS=RG(25)
      JSWPRN=TSTSWP
      PTRAP=RG(29)
C
      PI=3.141592653
      RPM=RG(830)
      SHP=RG(831)
      DIAFT=RG(52)/12.
      RHOAMB=RG(701)
      PCTK=RG(832)
      XOPROP=RG(43)*RG(31)
      YOPROP=RG(44)*RG(31)
      CALL WRIT2R(' XOPROP ',XOPROP,',YOPROP ',YOPROP)
C
      JCURVE=IG(875) + 1
      NRAMP=MAX0(IG(876),1)
C
      JNXNY=NX*NY
      JNXYZ=NX*NY*NZ
      RETURN
 1002 CONTINUE
      WRITE(6,175) JNXYZ
C
C ... CONVERT TO MKS UNITS ....
C
      CALL WRITBL
      CALL WRIT4R('  RPM    ',RPM,',  SHP   ',SHP,',Dia,Ft.',DIAFT,
     &             ', % Ke   ',PCTK)
```

```
      RPS=RPM/60.
      DIAM=DIAFT*.3048
      POWER=SHP*745.7
      XCP=POWER/(RHOAMB*RPS**3*DIAM**5)
      DBAR=0.75*DIAM
      CALL WRIT4R('  RPS   ',RPS,', POWER ',POWER,',Dia,M. ',DIAM,
     &            '   Cp   ',XCP)
      IF(XCP.LT.0.1 .OR. XCP.GT.0.8)
     &  CALL WRIT40(' ... BIZARRE Cp VALUE ....                ')

C ... INTERPOLATE TO GET Ct/Cp Vs. Cp ....
C
      XX=XCP
      DO 10020 II=2,NDATA
10020 IF(XX.LT.CTDATA(II,1)) GO TO 10025
      CALL WRIT40('ERROR IN INTERPOLATION FOR Ct/Cp Vs. Cp.')
      CALL WAYOUT(1)
10025 CONTINUE
      I0=II-1
      X0=CTDATA(I0  ,1)
      XP=CTDATA(I0+1,1)
      IF((XX-X0)/(XP-X0).GT.0.5) I0=I0+1

      X0=CTDATA(I0  ,1)
      XM=CTDATA(I0-1,1)
      XP=CTDATA(I0+1,1)
      Y0=CTDATA(I0  ,JCURVE)
      YM=CTDATA(I0-1,JCURVE)
      YP=CTDATA(I0+1,JCURVE)
      DX=(XP-XM)/2.
      YP0=(YP-YM)/(XP-XM)
      YPP0=(YM-2.*Y0+YP)/DX**2
      DX=XX-X0
      TERM1=YP0*DX
      TERM2=0.5*YPP0*DX**2
      YX1=Y0 + TERM1
      YX2=YX1 + TERM2
      CTBCP=YX2
      CALL WRITBL
      CALL WRIT40('Y = Ct/Cp IN THE FOLLOWING ....          ')
      CALL WRIT3R('  Cp,-   ',XM,', Cp,0   ',X0,', Cp,+   ',XP)
      CALL WRIT3R(' Ct/Cp,-',YM,',Ct/Cp,0',Y0,',Ct/Cp,+',YP)
      CALL WRIT4R('  TERM1 ',TERM1,', TERM2 ',TERM2,
     &            '   Y'',0 ',YP0,', Y'''',0 ',YPP0)
      CALL WRIT4R('   Cp   ',XX,',Y,O(1) ',YX1,',Y,O(2) ',YX2,
     &            ',Ct/Cp  ',CTBCP)
      THRUST=CTBCP*POWER/(RPS*DIAM)
      APROP=PI*DIAM**2/4.
      WPROP=SQRT(THRUST/(2.*RHOAMB*APROP))
      UPROP=POWER/(RHOAMB*APROP*WPROP*PI*RPS*DBAR)
      POWERW=THRUST*WPROP
      POWERK=PCTK/100.*POWER
      POWERU=POWER-POWERW-POWERK
      CALL WRIT4R(' THRUST ',THRUST,', Area  ',APROP,
     &            ',W,prop ',WPROP ,',U,prop ',UPROP)
      CALL WRIT4R('  Power ',POWER ,',Power,w',POWERW,
     &            ',Power,u',POWERU,',Power,k',POWERK)
C
      WPROP2=WPROP**2
      TBA=THRUST/APROP
      PUBA=POWERU/APROP
      PKBA=POWERK/APROP
      A2PI=2.*PI
      OMEGA=RPS*A2PI
C
      IF(JNY.GE.NY.AND.JNXY.GE.JNXNY) RETURN
      CALL WRIT3I('    NX    ',NX ,',   NY    ',NY ,', NXNY   ',JNXNY)
      CALL WRIT3I('    JNX   ',JNX,',   JNY   ',JNY,', NXNYD ',JNXY )
      WRITE(6,179)
      STOP
  175 FORMAT(/,1X,'TOTAL # CELLS :',I6)
  179 FORMAT(1X,'INCREASE JNX AND/OR JNY !!!!!',/,
     & 1X,'THE SHIT WOULD HAVE HIT THE FAN ...... STOPPING.')
C******************************************************************
C
C--- GROUP 2. Transience; time-step specification
C
    2 CONTINUE
      RETURN
C******************************************************************
C
C--- GROUP 3. X-direction grid specification
C
    3 CONTINUE
      RETURN
C******************************************************************
C
C--- GROUP 4. Y-direction grid specification
C
    4 CONTINUE
      RETURN
C******************************************************************
C
C--- GROUP 5. Z-direction grid specification
C
    5 CONTINUE
      RETURN
C******************************************************************
C
C--- GROUP 6. Body-fitted coordinates or grid distortion
C
    6 CONTINUE
      RETURN
C******************************************************************
C   * Make changes for this group only in group 19.
C--- GROUP 7. Variables stored, solved & named
C******************************************************************
C
C--- GROUP 8. Terms (in differential equations) & devices
C
    8 GO TO (81,82,83,84,85,86,87,88,89,810,811,812,813,814,815)
     1,ISC
   81 CONTINUE
C   * ------------------- SECTION  1 -------------------------
C    For U1AD.LE.GRND--- phase 1 additional velocity (VELAD).
      RETURN
   82 CONTINUE
C   * ------------------- SECTION  2 -------------------------
C    For U2AD.LE.GRND--- phase 2 additional velocity (VELAD).
      RETURN
   83 CONTINUE
C    * ------------------- SECTION  3 --------------------------
C     For V1AD.LE.GRND--- phase 1 additional velocity (VELAD).
      RETURN
   84 CONTINUE
C    * ------------------- SECTION  4 --------------------------
C     For V2AD.LE.GRND--- phase 2 additional velocity (VELAD).
      RETURN
   85 CONTINUE
C    * ------------------- SECTION  5 --------------------------
C     For W1AD.LE.GRND--- phase 1 additional velocity (VELAD).
      RETURN
   86 CONTINUE
C    * ------------------- SECTION  6 --------------------------
C     For W2AD.LE.GRND--- phase 2 additional velocity (VELAD).
      RETURN
   87 CONTINUE
C    * ------------------- SECTION 7 ---- VOLUMETRIC SOURCE FOR GALA
      RETURN
   88 CONTINUE
C    * ------------------- SECTION 8 --- CONVECTION FLUXES
      RETURN
   89 CONTINUE
C    * ------------------- SECTION 9 --- DIFFUSION COEFFICIENTS
      RETURN
  810 CONTINUE
C    * ------------------- SECTION 10 --- CONVECTION NEIGHBOURS
      RETURN
  811 CONTINUE
C    * ------------------- SECTION 11 --- DIFFUSION NEIGHBOURS
      RETURN
  812 CONTINUE
C    * ------------------- SECTION 12 --- LINEARISED SOURCES
      RETURN
  813 CONTINUE
C    * ------------------- SECTION 13 --- CORRECTION COEFFICIENTS
      RETURN
  814 CONTINUE
C    * ------------------- SECTION 14 --- USER'S SOLVER
      RETURN
  815 CONTINUE
C    * ------------------- SECTION 15 --- CHANGE SOLUTION
      RETURN
     * Make all other group-8 changes in group 19.
C*****************************************************************
C
C --- GROUP 9. Properties of the medium (or media)
C
C   The sections in this group are arranged sequentially in their
    order of calling from EARTH. Thus, as can be seen from below,
C   the temperature sections (10 and 11) precede the density
C   sections (1 and 3); so, density formulae can refer to
    temperature stores already set.
    9 GO TO (91,92,93,94,95,96,97,98,99,900,901,902,903),ISC
C*****************************************************************
  900 CONTINUE
     * ------------------- SECTION 10 --------------------------
C     For TMP1.LE.GRND--------- phase-1 temperature Index AUX(TEMP1)
      RETURN
  901 CONTINUE
```

```
C   * ------------------- SECTION 11 --------------------------
C    For TMP2.LE.GRND--------- phase-2 temperature Index AUX(TEMP2)
      RETURN
  902 CONTINUE
C   * ------------------- SECTION 12 --------------------------
C    For EL1.LE.GRND--------- phase-1 length scale Index AUX(LEN1)
      RETURN
  903 CONTINUE
C   * ------------------- SECTION 13 --------------------------
C    For EL2.LE.GRND--------- phase-2 length scale Index AUX(LEN2)
      RETURN
   91 CONTINUE
C   * ------------------- SECTION  1 --------------------------
C    For RHO1.LE.GRND--- density for phase 1 Index AUX(DEN1).
C
      CALL GETYX (P1,GP1,JNY,JNX)
      CALL GETYX (H1,GH1,JNY,JNX)
      CALL GETYX (C1,GC1,JNY,JNX)
      CALL GETYX (TEMP,GTMP,JNY,JNX)
      CALL GETYX (INAME('VPOR'),GVPR,JNY,JNX)
      DO 9101 IX=1,NX
      DO 9101 IY=1,NY
      IF (GVPR(IY,IX).LE.1.E-4) THEN
        GC3(IY,IX)=0.0
        GTMP(IY,IX)=300.
        PHI(IY,IX)=0.0
        GRH(IY,IX)=1.
        GCP(IY,IX)=1000.
        GOTO 9101
      ENDIF
C-pd---Calculate mass fractions------------------------------------
      GC3(IY,IX)=1.0-GC1(IY,IX)
      SC(1)=(GC3(IY,IX)*RG(1)+GC1(IY,IX)*RG(5))/RG(21)
      SC(2)=(GC3(IY,IX)*RG(2)+GC1(IY,IX)*RG(6))/RG(22)
      SC(3)=(GC3(IY,IX)*RG(3)+GC1(IY,IX)*RG(7))/RG(23)
      SC(4)=(GC3(IY,IX)*RG(4)+GC1(IY,IX)*RG(8))/RG(24)
      SC(1)=AMAX1(1.E-10,SC(1))
      SC(2)=AMAX1(1.E-10,SC(2))
      SC(3)=AMAX1(1.E-10,SC(3))
      SC(4)=AMAX1(1.E-10,SC(4))
      TGUS=GTMP(IY,IX)
      HSTAT=GH1(IY,IX)
      CALL TEMPER(HSTAT,TGUS,TCELL,CPDR,RGAS,SC,NSC,NFO)
C
      TCELL=AMAX1(VARMIN(TEMP),TCELL)
      TCELL=AMIN1(VARMAX(TEMP),TCELL)
C
      GP=PRESS0+GP1(IY,IX)
      PHI(IY,IX)=1.0/(GP+TNY)
      XMWA=1.0/(SC(1)+SC(2)+SC(3)+SC(4))
      GRH(IY,IX)=GP*XMWA/(RGAS*TCELL+TNY)
      GTMP(IY,IX)=TCELL
      GCP(IY,IX)=CPDR*RGAS
 9101 CONTINUE
C
      CALL SETYX(AUX(DEN1),GRH,JNY,JNX)
      CALL SETYX(C3,GC3,JNY,JNX)
      CALL SETYX(TEMP,GTMP,JNY,JNX)
      CALL SETYX(CP,GCP,JNY,JNX)
C
```

```
      RETURN
   92 CONTINUE
C   * ------------------- SECTION  2 ---------------------------
C    For DRH1DP.LE.GRND--- D(LN(DEN))/DP for phase 1 (D1DP).
      CALL SETYX(D1DP,PHI,JNY,JNX)
      RETURN
   93 CONTINUE
C   * ------------------- SECTION  3 ---------------------------
C    For RHO2.LE.GRND--- density for phase 2 Index AUX(DEN2).
      RETURN
   94 CONTINUE
C   * ------------------- SECTION  4 ---------------------------
C    For DRH2DP.LE.GRND--- D(LN(DEN))/DP for phase 2 (D2DP).
      RETURN
   95 CONTINUE
C   * ------------------- SECTION  5 ---------------------------
C    For ENUT.LE.GRND--- reference turbulent kinematic viscosity.
      RETURN
   96 CONTINUE
C   * ------------------- SECTION  6 ---------------------------
C    For ENUL.LE.GRND--- reference laminar kinematic viscosity.
      RETURN
   97 CONTINUE
C   * ------------------- SECTION  7 ---------------------------
C    For PRNDTL( ).LE.GRND--- laminar PRANDTL nos., or diffusivity.
      RETURN
   98 CONTINUE
C   * ------------------- SECTION  8 ---------------------------
C    For PHINT( ).LE.GRND--- interface value of first phase(FII1).
      RETURN
   99 CONTINUE
C   * ------------------- SECTION  9 ---------------------------
C    For PHINT( ).LE.GRND--- interface value of second phase(FII2)
      RETURN
C*****************************************************************
C
C--- GROUP 10. Inter-phase-transfer processes and properties
C
   10 GO TO (101,102,103,104),ISC
  101 CONTINUE
C   * ------------------- SECTION  1 ---------------------------
C    For CFIPS.LE.GRND--- inter-phase friction coeff. AUX(INTFRC).
      RETURN
  102 CONTINUE
C   * ------------------- SECTION  2 ---------------------------
C    For CMDOT.EQ.GRND- inter-phase mass transfer Index AUX(INTMDT)
      RETURN
  103 CONTINUE
C   * ------------------- SECTION  3 ---------------------------
C    For CINT( ).EQ.GRND--- phase1-to-interface transfer
C                                          coefficients (COI1)
      RETURN
  104 CONTINUE
C   * ------------------- SECTION  4 ---------------------------
C    For CINT( ).EQ.GRND--- phase2-to-interface transfer
C                                          coefficients (COI2)
      RETURN
C*****************************************************************
C
C--- GROUP 11. Initialization of variable or porosity fields
```

```
C
   11 CONTINUE
      RETURN
C****************************************************************
C
C--- GROUP 12. Convection and diffusion adjustments
C
   12 CONTINUE
      RETURN
C****************************************************************
C
C--- GROUP 13. Boundary conditions and special sources
C
   13 CONTINUE
      GO TO (130,131,132,133,134,135,136,137,138,139,1310,
     11311,1312,1313,1314,1315,1316,1317,1318,1319,1320,1321),ISC
  130 CONTINUE
C------------------- SECTION  1 ------------- coefficient = GRND
      RETURN
  131 CONTINUE
C------------------- SECTION  2 ------------- coefficient = GRND1
      RETURN
  132 CONTINUE
C------------------- SECTION  3 ------------- coefficient = GRND2
      RETURN
  133 CONTINUE
C------------------- SECTION  4 ------------- coefficient = GRND3
      RETURN
  134 CONTINUE
C------------------- SECTION  5 ------------- coefficient = GRND4
      RETURN
  135 CONTINUE
C------------------- SECTION  6 ------------- coefficient = GRND5
      RETURN
  136 CONTINUE
C------------------- SECTION  7 ------------- coefficient = GRND6
      RETURN
  137 CONTINUE
C------------------- SECTION  8 ------------- coefficient = GRND7
      IF(INDVAR.GT.P1) GO TO 13799
      CALL GETYX(AUX(DEN1),A1,NY,NX)
      CALL GETYX(P1       ,A2,NY,NX)
      CALL GETCOV(NPATCH,INAME('UCRT'),COEFF,GKLOSS)
      CALL GETCOV(NPATCH, P1 ,COEFF,GPBV  )
      I=(IXF-2)*NY
      DO 13701 II=IXF,IXL
        I=I + NY
        DO 13702 J=IYF,IYL
          IJ=I + J
          DELTAP=AMAX1(ABS(A2(IJ)-GPBV),PTRAP)
          RHO   =A1(IJ)
          COEFF =SQRT(2.*RHO/(GKLOSS*DELTAP))
          A1(IJ)=COEFF
13702   CONTINUE
13701   CONTINUE
      CALL SETYX(CO,A1,NY,NX)
      RETURN
13799 CALL WRIT40('CO = GRND7 FOR VARIABLE BESIDES P1 !!!! ')
      CALL WAYOUT(1)
      RETURN
  138 CONTINUE
C------------------- SECTION  9 ------------- coefficient = GRND8
C
C ... GENERATE WALL SHEAR COEFFICIENTS ....
C
      CALL FNLGLW(CO,CO,AK,1.0001,EWAL,4)
C
C ... NOW CONVERT TO Stanton #'s ....
C
      CALL GETYX(CO,A1,NY,NX)
      RPRL=1./PRNDTL(H1)
      RPRT=1./PRT(H1)
      P   =9.*(RPRT/RPRL - 1.)*(RPRL/RPRT)**0.25
      I=(IXF-2)*NY
      DO 13801 II=IXF,IXL
        I=I + NY
        DO 13802 J=IYF,IYL
          IJ=I + J
          S=A1(IJ)
          STL=S*RPRL
          STT=S*RPRT/(1. + P*SQRT(S))
          A1(IJ)=AMAX1(STL,STT)
.3802   CONTINUE
.3801 CONTINUE
C
C ... NOW ASSEMBLE COMPOSITE HEAT TRANSFER COEFFICIENTS ....
C
      CALL GETYX(AUX(DEN1),A2,NY,NX)
      CALL GETYX(LD7      ,A3,NY,NX)
      CALL GETYX(CP       ,A4,NY,NX)
      CALL GETCOV(NPATCH,INAME('UCRT'),COND,THICK)
      CWALL=COND/(THICK+TINY)
C
      CALL SUB4(I1,IXF,I2,IXL,J1,IYF,J2,IYL)
      READ(NPATCH(8:8),'(A1)') ADIR
      NDIREC=0
      IF(ADIR.EQ.'E' .OR. ADIR.EQ.'e') NDIREC= 1
      IF(ADIR.EQ.'W' .OR. ADIR.EQ.'w') NDIREC=-1
      IF(ADIR.EQ.'N' .OR. ADIR.EQ.'n') NDIREC= 2
      IF(ADIR.EQ.'S' .OR. ADIR.EQ.'s') NDIREC=-2
      IDIR=IABS(NDIREC)
      IF(IDIR.EQ.1) THEN
        KAREA=5
        KADD=NY
        I2=I1
      ELSEIF(IDIR.EQ.2) THEN
        KAREA=7
        KADD=1
        J2=J1
      ELSE
        CALL WRIT40('PATCH NAME PROTOCOL VIOLATED FOR GRND8   ')
        CALL WRIT40('COEFFICIENT OF CONJUGATE HEAT TRANSFER   ')
        CALL WRIT40('MODEL. TSK TSK TSK                       ')
        CALL WAYOUT(1)
      ENDIF
C
      I=(I1-2)*NY
      DO 13811 II=I1,I2
        I=I + NY
        DO 13812 J=J1,J2
```

```
          IJ1=I + J
          IJ2=IJ1 + KADD
          ST1=A1(IJ1)
          ST2=A1(IJ2)
          RO1=A2(IJ1)
          RO2=A2(IJ2)
          VW1=A3(IJ1)
          VW2=A3(IJ2)
          CP1=A4(IJ1)
          CP2=A4(IJ2)
          CO1=RO1*VW1*CP1*ST1
          CO2=RO2*VW2*CP2*ST2
          COEFF=CO1*CWALL*CO2/(CO1*CWALL + CO1*CO2 + CWALL*CO2 + TINY)
          A5(IJ1)=COEFF/CP1
          A5(IJ2)=COEFF/CP2
          A6(IJ1)=COEFF
          A6(IJ2)=COEFF
13812   CONTINUE
13811 CONTINUE
C
      CALL SETYX(C6,A5,NY,NX)
      CALL SETYX(C7,A6,NY,NX)
C
C ... NOW MULTIPLY BY CORRECT AREA'S & DIVIDE BY PATGEO,RHO & Vwall ....
C
      CALL GTIZYX(KAREA,IZ,A1,NY,NX)
      I=(I1-2)*NY
      DO 13821 II=I1,I2
        I=I + NY
        DO 13822 J=J1,J2
          IJ1=I + J
          IJ2=IJ1 + KADD
          AREA=A1(IJ1)
          A5(IJ1)=A5(IJ1)*AREA
          A5(IJ2)=A5(IJ2)*AREA
13822   CONTINUE
13821 CONTINUE
C
      CALL GETYX(PATGEO,A1,NY,NX)
      I=(IXF-2)*NY
      DO 13831 II=IXF,IXL
        I=I + NY
        DO 13832 J=IYF,IYL
          IJ=I + J
          A5(IJ)=A5(IJ)/(A1(IJ)*A2(IJ)*A3(IJ) + TINY)
13832   CONTINUE
13831 CONTINUE
C
      CALL SETYX(CO,A5,NY,NX)
      CALL FN1(LGEN1,0.0)
C
C ... ADD UP TOTAL HEAT TRANSFERRED ....
C
      IF(ISWEEP.LT.LSWEEP-1.AND.MOD(ISWEEP,IG(901)).NE.0) RETURN
C
C     CALL WRITBL
C     CALL WRIT40('ADDING UP TOTAL Qdot FROM DUCT TO AIR.  ')
C     CALL WRIT2I('SWEEP # ',ISWEEP,',SLAB # ',IZSTEP)
      CALL GETYX(H1,A4,NY,NX)
      CALL GETYX(CP,A2,NY,NX)
      CALL SUB4(I1,IXF,I2,IXL,J1,IYF,J2,IYL)
      IF    (NDIREC.EQ. 1) THEN
        I1=I2
        KADD=-NY
      ELSEIF(NDIREC.EQ.-1) THEN
        I2=I1
        KADD= NY
      ELSEIF(NDIREC.EQ. 2) THEN
        J1=J2
        KADD=-1
      ELSEIF(NDIREC.EQ.-2) THEN
        J2=J1
        KADD= 1
      ENDIF
C
      READ(NPATCH(7:7),'(A1)') ANUX
      I=(I1-2)*NY
      DO 13841 II=I1,I2
        I=I + NY
        DO 13842 J=J1,J2
          IJ1=I + J
          IJ2=IJ1 + KADD
          H11=A4(IJ1)
          H12=A4(IJ2)
          CP1=A2(IJ1)
          CP2=A2(IJ2)
          CO1=A5(IJ1)*A1(IJ1)*A3(IJ1)
          VA1=H12*CP1/CP2
          QDTTOT=QDTTOT + CO1*(VA1-H11)
          IF(ANUX.EQ.'1') QDOT01=QDOT01 + CO1*(VA1-H11)
          IF(ANUX.EQ.'2') QDOT02=QDOT02 + CO1*(VA1-H11)
          IF(ANUX.EQ.'3') QDOT03=QDOT03 + CO1*(VA1-H11)
          IF(ANUX.EQ.'4') QDOT04=QDOT04 + CO1*(VA1-H11)
13842   CONTINUE
13841 CONTINUE
C
      RETURN
  139 CONTINUE
C------------------- SECTION 10 ------------- coefficient = GRND9
      RETURN
 1310 CONTINUE
C------------------- SECTION 11 ------------- coefficient = GRND10
      RETURN
 1311 CONTINUE
C------------------- SECTION 12 -------------------- value = GRND
      RETURN
 1312 CONTINUE
C------------------- SECTION 13 -------------------- value = GRND1
      RETURN
 1313 CONTINUE
C------------------- SECTION 14 -------------------- value = GRND2
      RETURN
 1314 CONTINUE
C------------------- SECTION 15 -------------------- value = GRND3
      RETURN
 1315 CONTINUE
C------------------- SECTION 16 -------------------- value = GRND4
      RETURN
 1316 CONTINUE
C------------------- SECTION 17 -------------------- value = GRND5
```

```
      RETURN
 1317 CONTINUE
C------------------- SECTION 18 -------------------- value = GRND6
      RETURN
 1318 CONTINUE
C------------------- SECTION 19 -------------------- value = GRND7
      IF(INDVAR.LT.U1 .OR. INDVAR.GT.W2) GO TO 13189
      CALL GETYX(AUX(DEN1),A1,NY,NX)
      CALL GETYX(P1        ,A2,NY,NX)
      CALL GETCOV(NPATCH,INAME('UCRT'),COEFF,GKLOSS)
      CALL GETCOV(NPATCH, P1 ,COEFF,GPBV   )
      I=(IXF-2)*NY
      DO 13181 II=IXF,IXL
        I=I + NY
        DO 13182 J=IYF,IYL
          IJ=I + J
          DELTAP= A2(IJ)-GPBV
          ABSDP = ABS(DELTAP)
          RHO   = A1(IJ)
          VMAG  = SQRT(2.*ABSDP/(GKLOSS*RHO))
          A1(IJ)=-SIGN(VMAG,DELTAP)
13182   CONTINUE
13181 CONTINUE
      CALL SETYX(VAL,A1,NY,NX)
      RETURN
13189 CALL WRIT40('VAL = GRND7 FOR VARBLE BESIDES ÄU,V,WÅ1.')
      CALL WAYOUT(1)
      RETURN
 1319 CONTINUE
C------------------- SECTION 20 -------------------- value = GRND8
      CALL GETYX(H1,A1,NY,NX)
      CALL GETYX(CP,A2,NY,NX)
C
      I=(I1-2)*NY
      DO 13191 II=I1,I2
        I=I + NY
        DO 13192 J=J1,J2
          IJ1=I + J
          IJ2=IJ1 + KADD
          H11=A1(IJ1)
          H12=A1(IJ2)
          CP1=A2(IJ1)
          CP2=A2(IJ2)
          VA1=H12*CP1/CP2
          VA2=H11*CP2/CP1
          A3(IJ1)=VA1
          A3(IJ2)=VA2
13192   CONTINUE
13191 CONTINUE
C
      CALL SETYX(VAL,A3,NY,NX)
      RETURN
 1320 CONTINUE
C------------------- SECTION 21 -------------------- value = GRND9
      IF(INDVAR.LT.W1) THEN
        IF(INDVAR.EQ.V1) THEN
          CALL GTIZYX(83,IZ,A1,NY,NX)
          CALL GTIZYX(84,IZ,A2,NY,NX)
          IF(LG(20))
     &      CALL WRIT40('IN GROUP 13,V1 SECTION ....                  ')
```

```
        I1=(IXF-2)*NY
        IJ1=I1+IYF+1
        IJ2=I1+IYL
        IF(LG(20))
    &     CALL WRIT40('BEGIN 1ST SOURCE LOOP ....                ')
        DO 13111 I=IXF,IXL
        IJ1=IJ1+NY
        IJ2=IJ2+NY
        IF(LG(20)) CALL WRIT3I('   IX    ', I ,', IJ1    ',IJ1,
    &                          ',  IJ2  ',IJ2)
        DO 13111 IJ=IJ1,IJ2
13111   A3(IJ)=0.5*FMAG(IJ)*(EFX(IJ)*A1(IJ) +
    &                              EFY(IJ)*A2(IJ))
        I1=(IXF-2)*NY
        IJ1=I1+IYF
        IJ2=I1+IYL-1
        IF(LG(20))
    &     CALL WRIT40('BEGIN 2ND SOURCE LOOP ....                ')
        DO 13112 I=IXF,IXL
        IJ1=IJ1+NY
        IJ2=IJ2+NY
        IF(LG(20)) CALL WRIT3I('   IX    ', I ,', IJ1    ',IJ1,
    &                          ',  IJ2  ',IJ2)
        DO 13112 IJ=IJ1,IJ2
13112   A3(IJ)=A3(IJ) + 0.5*FMAG(IJ+1)*(EFX(IJ+1)*A1(IJ) +
    &                                          EFY(IJ+1)*A2(IJ))
        CALL SETYX(VAL,A3,NY,NX)
        RETURN
      ENDIF
      IF(LG(20))
    &   CALL WRIT40('IN GROUP 13,U1 SECTION ....               ')
      CALL GTIZYX(80,IZ,A1,NY,NX)
      CALL GTIZYX(81,IZ,A2,NY,NX)
      I1=(IXF-1)*NY
      IJ1=I1+IYF
      IJ2=I1+IYL
      DO 13113 I=IXF,IXL
      IJ1=IJ1+NY
      IJ2=IJ2+NY
      IF(LG(20)) CALL WRIT3I('   IX    ', I ,', IJ1    ',IJ1,
    &                        ',  IJ2  ',IJ2)
      DO 13113 IJ=IJ1,IJ2
13113 A3(IJ)=0.5*FMAG(IJ)*(EFX(IJ)*A1(IJ) +
    &                            EFY(IJ)*A2(IJ))
      I1=(IXF-2)*NY
      IJ1=I1+IYF
      IJ2=I1+IYL
      DO 13114 I=IXF,IXL-1
      IJ1=IJ1+NY
      IJ2=IJ2+NY
      IF(LG(20)) CALL WRIT3I('   IX    ', I ,', IJ1    ',IJ1,
    &                        ',  IJ2  ',IJ2)
      DO 13114 IJ=IJ1,IJ2
13114 A3(IJ)=A3(IJ) + 0.5*FMAG(IJ+NY)*(EFX(IJ+NY)*A1(IJ) +
    &                                        EFY(IJ+NY)*A2(IJ))
      CALL SETYX(VAL,A3,NY,NX)
      RETURN
C
C ... W1 SECTION ....
C
```

```
      ELSEIF(INDVAR.EQ.W1) THEN
        IF(LG(20))
     &      CALL WRIT40('IN GROUP 13,W1 SECTION ....                ')
        CALL GTIZYX(28,IZ,A1,NY,NX)
        CONST=2.*WPROP2
        DO 13115 IJ=1,JNXNY
13115   A1(IJ)=CONST/(A1(IJ) + TINY)
        CALL SETYX(VAL,A1,NY,NX)
        RETURN
C
C ... KE-EP SECTION ....
C
C ... Pk(r)=Ck * Uprop(r)**2  W/Ck=64*POWERK/(2*Pi * OMEGA**2 * DIAM**4)
C     Pk(r)=CK * (OMEGA*r)**2
C
      ELSE
        FRAC=(FLOAT(ISWEEP-FSWEEP+1)/FLOAT(NRAMP))**2
        FRAC=AMIN1(AMAX1(0.,FRAC),1.)
        CK = FRAC * 64.*POWERK/(A2PI * OMEGA**2 * DIAM**4)
        I1=(IXF-2)*NY
        IJ1=I1+IYF
        IJ2=I1+IYL
        IF(LG(20))
     &    CALL WRIT40('BEGIN Pk(r) LOOP ....                    ')
        DO 13118 I=IXF,IXL
        IJ1=IJ1+NY
        IJ2=IJ2+NY
        IF(LG(20)) CALL WRIT3I('   IX    ',.I ,', IJ1    ',IJ1,
     &                         ',  IJ2   ',IJ2)        .
        DO 13118 IJ=IJ1,IJ2
          RR=RP(IJ)
          RW=RR*OMEGA
13118     A1(IJ)=CK*RW**2
        IF(INDVAR.GT.KE) THEN
          CALL GETYX(AUX(VIST),A2,NY,NX)
          CALL GETYX(AUX(LEN1),A3,NY,NX)
          CONST=C1E*CD**2/CMUCD
          I1=(IXF-2)*NY
          IJ1=I1+IYF
          IJ2=I1+IYL
          IF(LG(20))
     &      CALL WRIT40('BEGIN C1*Pk(r)*EP/KE LOOP ....          ')
          DO 13119 I=IXF,IXL
          IJ1=IJ1+NY
          IJ2=IJ2+NY
          IF(LG(20)) CALL WRIT3I('   IX    ', I ,', IJ1    ',IJ1,
     &                           ',  IJ2   ',IJ2)
          DO 13119 IJ=IJ1,IJ2
13119     A1(IJ)=CONST*A1(IJ)*A2(IJ)/A3(IJ)**2
        ENDIF
        CALL SETYX(VAL,A1,NY,NX)
      ENDIF
      RETURN
 1321 CONTINUE
C------------------- SECTION 22 ------------------- value = GRND10
      DO 13211 IX=1,NX
      DO 13211 IY=1,NY
      PHI(IY,IX)=RG(804)*XFCTE
      IF(NPATCH.EQ.'XENGIN') PHI(IY,IX)=-RG(804)*XFCTE
      IF(NPATCH.EQ.'XENGOUT') PHI(IY,IX)=RG(805)*XFCTE
```

```
13211 CONTINUE
      CALL SETYX(VAL,PHI,JNY,JNX)
      RETURN
C***************************************************************
C
C--- GROUP 14. Downstream pressure for PARAB=.TRUE.
C
   14 CONTINUE
      RETURN
C***************************************************************
C   * Make changes for this group only in group 19.
C--- GROUP 15. Termination of sweeps
C--- GROUP 16. Termination of iterations
C--- GROUP 17. Under-relaxation devices
C--- GROUP 18. Limits on variables or increments to them
C***************************************************************
C
C--- GROUP 19. Special calls to GROUND from EARTH
C
   19 GO TO (191,192,193,194,195,196,197,198),ISC
  191 CONTINUE
C   * ------------------- SECTION 1 ---- START OF TIME STEP.
C
C-pd---Misc----------------------------------------------------------
C
      IF(IG(999).EQ.1) STOP
      QDTTOT=0.0
      QDOTT1=0.0
      QDOTT2=0.0
      QDOTT3=0.0
      QDOTT4=0.0
      IPASS=0
      IRAXV=0
      IRAXT=0
      IRAXS=0
      XFCTE=1.0
      ITST=TSTSWP
      INPR=NPRMON
      NPRMON=1
C
C-pd---Assign monitoring locations-----------------------------------
C
      IXMON1 =IXMON
      IYMON1 =IYMON
      IZMON1 =IZMON
C
      IXMON2 =IG(11)
      IYMON2 =IG(12)
      IZMON2 =IG(13)
C
      IXMON3 =IG(14)
      IYMON3 =IG(15)
      IZMON3 =IG(16)
C
      IXMON4 =IG(17)
      IYMON4 =IG(18)
      IZMON4 =IG(19)
C
      IXMON5 =IG(20)
      IYMON5 =IG(21)
      IZMON5 =IG(22)
C
      IXMON6 =IG(23)
      IYMON6 =IG(24)
      IZMON6 =IG(25)
C
      IXMON7 =IG(26)
      IYMON7 =IG(27)
      IZMON7 =IG(28)
C
      IXMON8 =IG(29)
      IYMON8 =IG(30)
      IZMON8 =IG(31)
C
      IXMON9 =IG(32)
      IYMON9 =IG(33)
      IZMON9 =IG(34)
C
      IXMON10=IG(35)
      IYMON10=IG(36)
      IZMON10=IG(37)
C
      RETURN
  192 CONTINUE
C   * ------------------- SECTION 2 ---- START OF SWEEP.
C
C-pd---WARNING:  machine dependent------------------------------------
C
C     call flush(6)
C
C ... COMPUTE rP,Fmag,eFx,eFy ....
C
      IF(ISWEEP.EQ.FSWEEP) THEN
        CALL GETPTC('YPROP   ',TYPE,IXF,IXL,IYF,IYL,IZF,IZL,ITF,ITL)
        CALL GTIZYX(68,IZF,A1,NY,NX)
        CALL GTIZYX(69,IZF,A2,NY,NX)
        EWZ=SIGN(1.,RPS)
        I1=(IXF-2)*NY
        IJ1=I1+IYF+1
        IJ2=I1+IYL
        IF(LG(20))
     &    CALL WRIT40('BEGIN Fmag LOOP ....                    ')
        DO 19201 I=IXF,IXL
        IJ1=IJ1+NY
        IJ2=IJ2+NY
        IF(LG(20)) CALL WRIT3I('   IX    ', I ,', IJ1    ',IJ1,
     &                         ',  IJ2   ',IJ2)
        DO 19201 IJ=IJ1,IJ2
        RX=A1(IJ)-XOPROP
        RY=A2(IJ)-YOPROP
        RR=SQRT(RX*RX + RY*RY) + TINY
        RP(IJ)=RR
        RW=RR*OMEGA
        ERX=RX/RR
        ERY=RY/RR
        EFX(IJ)=-EWZ*ERY
        EFY(IJ)= EWZ*ERX
19201   FMAG(IJ)=PUBA/RW
        IF((LSWEEP-FSWEEP).LT.11) THEN
          CALL SUB4(IXF,1,IXL,NX,IYF,1,IYL,NY)
          CALL WRIT2R(' X0,prop',X0PROP,',Y0,prop',Y0PROP)
          CALL PRNYX(' rP ',RP  ,NY,NX)
          CALL PRNYX(' eFx',EFX ,NY,NX)
          CALL PRNYX(' eFy',EFY ,NY,NX)
          CALL PRNYX(' öFö',FMAG,NY,NX)
        ENDIF
      ENDIF
C
C-pd---Check to reset tstswp---------------------------------------
C
      IOPEN=0
      IF(ITST.NE.TSTSWP) IPASS=IPASS+1
      IF(IPASS.GT.10) THEN
        IPASS=0
        TSTSWP=ITST
      ENDIF
C
C-pd---Init stuff for printout of max and min-----------------------
C
      XP1MIN= 1000000.0
      XP1MAX=-1000000.0
      XU1MIN= 1000000.0
      XU1MAX=-1000000.0
      XV1MIN= 1000000.0
      XV1MAX=-1000000.0
      XW1MIN= 1000000.0
      XW1MAX=-1000000.0
      XKEMIN= 1000000.0
      XKEMAX=-1000000.0
      XEPMIN= 1000000.0
      XEPMAX=-1000000.0
      XH1MIN= 1000000.0
      XH1MAX=-1000000.0
      XT1MIN= 1000000.0
      XT1MAX=-1000000.0
      XETMIN= 1000000.0
      XETMAX=-1000000.0
      IXPMAX=0
      IYPMAX=0
      IZPMAX=0
      IXPMIN=0
      IYPMIN=0
      IZPMIN=0
      IXUMAX=0
      IYUMAX=0
      IZUMAX=0
      IXUMIN=0
      IYUMIN=0
      IZUMIN=0
      IXVMAX=0
      IYVMAX=0
      IZVMAX=0
      IXVMIN=0
      IYVMIN=0
      IZVMIN=0
      IXWMAX=0
      IYWMAX=0
      IZWMAX=0
      IXWMIN=0
      IYWMIN=0
      IZWMIN=0
      IXKMAX=0
      IYKMAX=0
      IZKMAX=0
      IXKMIN=0
      IYKMIN=0
      IZKMIN=0
      IXEMAX=0
      IYEMAX=0
      IZEMAX=0
      IXEMIN=0
      IYEMIN=0
      IZEMIN=0
      IXHMAX=0
      IYHMAX=0
      IZHMAX=0
      IXHMIN=0
      IYHMIN=0
      IZHMIN=0
      IXTMAX=0
      IYTMAX=0
      IZTMAX=0
      IXTMIN=0
      IYTMIN=0
      IZTMIN=0
      IXXMAX=0
      IYXMAX=0
      IZXMAX=0
      IXXMIN=0
      IYXMIN=0
      IZXMIN=0
C
      RETURN
  193 CONTINUE
C    * ------------------- SECTION 3 ---- START OF IZ SLAB.
      RETURN
  194 CONTINUE
C    * ------------------- SECTION 4 ---- START OF ITERATION.
C     IF(IRAXV.EQ.1) THEN
C       CALL XSETCV('RAX1', U1,XCOF,XVEL,RAXFTV,1.0)
C       CALL XSETCV('RAX1', V1,XCOF,XVEL,RAXFTV,1.0)
C       CALL XSETCV('RAX1', W1,XCOF,XVEL,RAXFTV,1.0)
C       WRITE(6,*)' CO FROM SETCV VEL -> ',XCOF
C       IRAXV=0
C     ENDIF
C     IF(IRAXT.EQ.1) THEN
C       CALL XSETCV('RAX1', KE,XCOF,XVEL,RAXFTT,1.0)
C       CALL XSETCV('RAX1', EP,XCOF,XVEL,RAXFTT,1.0)
C       WRITE(6,*)' CO FROM SETCV TUR -> ',XCOF
C       IRAXT=0
C     ENDIF
C     IF(IRAXS.EQ.1) THEN
C       CALL XSETCV('RAX1', H1,XCOF,XVEL,RAXFTS,1.0)
C       CALL XSETCV('RAX1', C1,XCOF,XVEL,RAXFTS,1.0)
C       CALL XSETCV('RAX1', C2,XCOF,XVEL,RAXFTS,1.0)
C       WRITE(6,*)' CO FROM SETCV SCA -> ',XCOF
C       IRAXS=0
C      ENDIF
C
C-pd---Modify inlet areas--------------------------------------------
```

```
C
      IF(IZ.EQ.IG(711)) THEN
        CALL GTIZYX(9,IZ,GAH,JNY,JNX)
        SUMB=0.0
        DO 19302 IX=IG(712),IG(713)
        DO 19302 IY=IG(714),IG(715)
        SUMB=SUMB+GAH(IY,IX)
19302   CONTINUE
      ENDIF
C
      IF(IZ.EQ.NZ) THEN
        XFCTE=RG(802)/SUMB
C       CALL XSETCV('XENGOUT',P1,XCOF,XVEL,1.0,XFCTE)
C       CALL XSETCV('XENGIN', P1,XCOF,XVEL,1.0,XFCTE)
      ENDIF
C
      RETURN
  195 CONTINUE
C   * ------------------- SECTION 5 ---- FINISH OF ITERATION.
      RETURN
  196 CONTINUE
C   * ------------------- SECTION 6 ---- FINISH OF IZ SLAB.
      CALL GETCAR
      IF (MOD(ISWEEP,IG(902)).NE.0.AND.ISWEEP.NE.LSWEEP-1) GOTO 1961
      IF(IZ.EQ.1) WRITE(6,*)' ==> CALCULATING ENGLISH UNITS '
C-pd---Dispensed by DBS for unknown reasons??????????????????????????
C     CALL BCARTC(1,1)
C
      CALL GETYX(P1,PHI,JNY,JNX)
      DO 19611 IX=1,NX
      DO 19611 IY=1,NY
19611 PHI(IY,IX)=PHI(IY,IX)*RG(36)
      CALL SETYX(PH2O,PHI,JNY,JNX)
C
      CALL GETYX(INAME('UCRT'),PHI,JNY,JNX)
      DO 19612 IX=1,NX
      DO 19612 IY=1,NY
19612 PHI(IY,IX)=PHI(IY,IX)*RG(37)
      CALL SETYX(U2,PHI,JNY,JNX)
C
      CALL GETYX(INAME('VCRT'),PHI,JNY,JNX)
      DO 19613 IX=1,NX
      DO 19613 IY=1,NY
 9613 PHI(IY,IX)=PHI(IY,IX)*RG(37)
      CALL SETYX(V2,PHI,JNY,JNX)
C
      CALL GETYX(INAME('WCRT'),PHI,JNY,JNX)
      DO 19614 IX=1,NX
      DO 19614 IY=1,NY
 9614 PHI(IY,IX)=PHI(IY,IX)*RG(37)
      CALL SETYX(W2,PHI,JNY,JNX)
C
      CALL GETYX(TEMP,PHI,JNY,JNX)
      DO 19615 IX=1,NX
      DO 19615 IY=1,NY
19615 PHI(IY,IX)=PHI(IY,IX)/RG(33)-RG(32)
      CALL SETYX(TFAR,PHI,JNY,JNX)
C
      CALL GETYX(AUX(DEN1),PHI,JNY,JNX)
      DO 19616 IX=1,NX
      DO 19616 IY=1,NY
19616 PHI(IY,IX)=PHI(IY,IX)*RG(38)
      CALL SETYX(RHOE,PHI,JNY,JNX)
C
C-pd---Find max and min-----------------------------------------------
C
 1961 IF(MOD(ISWEEP,NPRMON).EQ.0) THEN
        CALL GETYX(P1,PHI,JNY,JNX)
        CALL GETYX (INAME('VPOR'),GVPR,JNY,JNX)
        DO 19617 IX=1,NX
        DO 19617 IY=1,NY
        IF (GVPR(IY,IX).LE.1.E-4) GOTO 19617
        IF(PHI(IY,IX).GT.XP1MAX) THEN
          XP1MAX=PHI(IY,IX)
          IXPMAX=IX
          IYPMAX=IY
          IZPMAX=IZ
        ENDIF
        IF(PHI(IY,IX).LT.XP1MIN) THEN
          XP1MIN=PHI(IY,IX)
          IXPMIN=IX
          IYPMIN=IY
          IZPMIN=IZ
        ENDIF
19617   CONTINUE
C
        CALL GETYX(U1,PHI,JNY,JNX)
        CALL GETYX (INAME('VPOR'),GVPR,JNY,JNX)
        DO 19618 IX=1,NX
        DO 19618 IY=1,NY
        IF (GVPR(IY,IX).LE.1.E-4) GOTO 19618
        IF(PHI(IY,IX).GT.XU1MAX) THEN
          XU1MAX=PHI(IY,IX)
          IXUMAX=IX
          IYUMAX=IY
          IZUMAX=IZ
        ENDIF
        IF(PHI(IY,IX).LT.XU1MIN) THEN
          XU1MIN=PHI(IY,IX)
          IXUMIN=IX
          IYUMIN=IY
          IZUMIN=IZ
        ENDIF
19618   CONTINUE
C
        CALL GETYX(V1,PHI,JNY,JNX)
        CALL GETYX (INAME('VPOR'),GVPR,JNY,JNX)
        DO 19619 IX=1,NX
        DO 19619 IY=1,NY
        IF (GVPR(IY,IX).LE.1.E-4) GOTO 19619
        IF(PHI(IY,IX).GT.XV1MAX) THEN
          XV1MAX=PHI(IY,IX)
          IXVMAX=IX
          IYVMAX=IY
          IZVMAX=IZ
        ENDIF
        IF(PHI(IY,IX).LT.XV1MIN) THEN
          XV1MIN=PHI(IY,IX)
          IXVMIN=IX
          IYVMIN=IY
          IZVMIN=IZ
        ENDIF
19619   CONTINUE
C
        CALL GETYX(W1,PHI,JNY,JNX)
        CALL GETYX (INAME('VPOR'),GVPR,JNY,JNX)
        DO 19620 IX=1,NX
        DO 19620 IY=1,NY
        IF (GVPR(IY,IX).LE.1.E-4) GOTO 19620
        IF(PHI(IY,IX).GT.XW1MAX) THEN
          XW1MAX=PHI(IY,IX)
          IXWMAX=IX
          IYWMAX=IY
          IZWMAX=IZ
        ENDIF
        IF(PHI(IY,IX).LT.XW1MIN) THEN
          XW1MIN=PHI(IY,IX)
          IXWMIN=IX
          IYWMIN=IY
          IZWMIN=IZ
        ENDIF
19620   CONTINUE
C
        CALL GETYX(KE,PHI,JNY,JNX)
        CALL GETYX (INAME('VPOR'),GVPR,JNY,JNX)
        DO 19621 IX=1,NX
        DO 19621 IY=1,NY
        IF (GVPR(IY,IX).LE.1.E-4) GOTO 19621
        IF(PHI(IY,IX).GT.XKEMAX) THEN
          XKEMAX=PHI(IY,IX)
          IXKMAX=IX
          IYKMAX=IY
          IZKMAX=IZ
        ENDIF
        IF(PHI(IY,IX).LT.XKEMIN) THEN
          XKEMIN=PHI(IY,IX)
          IXKMIN=IX
          IYKMIN=IY
          IZKMIN=IZ
        ENDIF
19621   CONTINUE
C
        CALL GETYX(EP,PHI,JNY,JNX)
        CALL GETYX (INAME('VPOR'),GVPR,JNY,JNX)
        DO 19622 IX=1,NX
        DO 19622 IY=1,NY
        IF (GVPR(IY,IX).LE.1.E-4) GOTO 19622
        IF(PHI(IY,IX).GT.XEPMAX) THEN
          XEPMAX=PHI(IY,IX)
          IXEMAX=IX
          IYEMAX=IY
          IZEMAX=IZ
        ENDIF
        IF(PHI(IY,IX).LT.XEPMIN) THEN
          XEPMIN=PHI(IY,IX)
          IXEMIN=IX
          IYEMIN=IY
          IZEMIN=IZ
        ENDIF
 9622   CONTINUE
```

```
C
        CALL GETYX(H1,PHI,JNY,JNX)
        CALL GETYX (INAME('VPOR'),GVPR,JNY,JNX)
        DO 19623 IX=1,NX
        DO 19623 IY=1,NY
        IF (GVPR(IY,IX).LE.1.E-4) GOTO 19623
        IF(PHI(IY,IX).GT.XH1MAX) THEN
          XH1MAX=PHI(IY,IX)
          IXHMAX=IX
          IYHMAX=IY
          IZHMAX=IZ
        ENDIF
        IF(PHI(IY,IX).LT.XH1MIN) THEN
          XH1MIN=PHI(IY,IX)
          IXHMIN=IX
          IYHMIN=IY
          IZHMIN=IZ
        ENDIF
19623   CONTINUE
C
        CALL GETYX(TEMP,PHI,JNY,JNX)
        CALL GETYX (INAME('VPOR'),GVPR,JNY,JNX)
        DO 19624 IX=1,NX
        DO 19624 IY=1,NY
        IF (GVPR(IY,IX).LE.1.E-4) GOTO 19624
        IF(PHI(IY,IX).GT.XT1MAX) THEN
          XT1MAX=PHI(IY,IX)
          IXTMAX=IX
          IYTMAX=IY
          IZTMAX=IZ
        ENDIF
        IF(PHI(IY,IX).LT.XT1MIN) THEN
          XT1MIN=PHI(IY,IX)
          IXTMIN=IX
          IYTMIN=IY
          IZTMIN=IZ
        ENDIF
19624   CONTINUE
C
        CALL GETYX(AUX(VIST),PHI,JNY,JNX)
        CALL GETYX (INAME('VPOR'),GVPR,JNY,JNX)
        DO 19625 IX=1,NX
        DO 19625 IY=1,NY
        IF (GVPR(IY,IX).LE.1.E-4) GOTO 19625
        IF(PHI(IY,IX).GT.XETMAX) THEN
          XETMAX=PHI(IY,IX)
          IXXMAX=IX
          IYXMAX=IY
          IZXMAX=IZ
        ENDIF
        IF(PHI(IY,IX).LT.XETMIN) THEN
          XETMIN=PHI(IY,IX)
          IXXMIN=IX
          IYXMIN=IY
          IZXMIN=IZ
        ENDIF
19625   CONTINUE
      ENDIF
C
C-pd---Get monitoring values------------------------------------------
```

```
C
      IF(MOD(ISWEEP,TSTSWP).NE.0) GOTO 19692
      IF(IZ.NE.IZMON1) GOTO 1962
        CALL GETONE(P1,PP1,IYMON1,IXMON1)
        CALL GETONE(U1,UU1,IYMON1,IXMON1)
        CALL GETONE(V1,VV1,IYMON1,IXMON1)
        CALL GETONE(W1,WW1,IYMON1,IXMON1)
        CALL GETONE(AUX(DEN1),DD1,IYMON1,IXMON1)
        IF(STORE(KE)) CALL GETONE(KE,KE1,IYMON1,IXMON1)
        IF(STORE(EP)) CALL GETONE(EP,EP1,IYMON1,IXMON1)
        IF(STORE(AUX(VIST))) CALL GETONE(AUX(VIST),ET1,IYMON1,IXMON1)
        CALL GETONE(C1,C1C1,IYMON1,IXMON1)
        IF(STORE(C2)) CALL GETONE(C2,C2C1,IYMON1,IXMON1)
        IF(STORE(C3)) CALL GETONE(C3,C3C1,IYMON1,IXMON1)
        IF(STORE(CP)) CALL GETONE(CP,CPC1,IYMON1,IXMON1)
        IF(STORE(C11)) CALL GETONE(C11,CXC1,IYMON1,IXMON1)
        CALL GETONE(TEMP,C4C1,IYMON1,IXMON1)
        CALL GETONE(H1,H1H1,IYMON1,IXMON1)
 1962 IF(IZ.NE.IZMON2) GOTO 1963
        CALL GETONE(P1,PP2,IYMON2,IXMON2)
        CALL GETONE(U1,UU2,IYMON2,IXMON2)
        CALL GETONE(V1,VV2,IYMON2,IXMON2)
        CALL GETONE(W1,WW2,IYMON2,IXMON2)
        CALL GETONE(AUX(DEN1),DD2,IYMON2,IXMON2)
        IF(STORE(KE)) CALL GETONE(KE,KE2,IYMON2,IXMON2)
        IF(STORE(EP)) CALL GETONE(EP,EP2,IYMON2,IXMON2)
        IF(STORE(AUX(VIST))) CALL GETONE(AUX(VIST),ET2,IYMON2,IXMON2)
        CALL GETONE(C1,C1C2,IYMON2,IXMON2)
        IF(STORE(C2)) CALL GETONE(C2,C2C2,IYMON2,IXMON2)
        IF(STORE(C3)) CALL GETONE(C3,C3C2,IYMON2,IXMON2)
        IF(STORE(CP)) CALL GETONE(CP,CPC2,IYMON2,IXMON2)
        IF(STORE(C11)) CALL GETONE(C11,CXC2,IYMON2,IXMON2)
        CALL GETONE(TEMP,C4C2,IYMON2,IXMON2)
        CALL GETONE(H1,H1H2,IYMON2,IXMON2)
 1963 IF(IZ.NE.IZMON3) GOTO 1964
        CALL GETONE(P1,PP3,IYMON3,IXMON3)
        CALL GETONE(U1,UU3,IYMON3,IXMON3)
        CALL GETONE(V1,VV3,IYMON3,IXMON3)
        CALL GETONE(W1,WW3,IYMON3,IXMON3)
        CALL GETONE(AUX(DEN1),DD3,IYMON3,IXMON3)
        IF(STORE(KE)) CALL GETONE(KE,KE3,IYMON3,IXMON3)
        IF(STORE(EP)) CALL GETONE(EP,EP3,IYMON3,IXMON3)
        IF(STORE(AUX(VIST))) CALL GETONE(AUX(VIST),ET3,IYMON3,IXMON3)
        CALL GETONE(C1,C1C3,IYMON3,IXMON3)
        IF(STORE(C2)) CALL GETONE(C2,C2C3,IYMON3,IXMON3)
        IF(STORE(C3)) CALL GETONE(C3,C3C3,IYMON3,IXMON3)
        IF(STORE(CP)) CALL GETONE(CP,CPC3,IYMON3,IXMON3)
        IF(STORE(C11)) CALL GETONE(C11,CXC3,IYMON3,IXMON3)
        CALL GETONE(TEMP,C4C3,IYMON3,IXMON3)
        CALL GETONE(H1,H1H3,IYMON3,IXMON3)
 1964 IF(IZ.NE.IZMON4) GOTO 1965
        CALL GETONE(P1,PP4,IYMON4,IXMON4)
        CALL GETONE(U1,UU4,IYMON4,IXMON4)
        CALL GETONE(V1,VV4,IYMON4,IXMON4)
        CALL GETONE(W1,WW4,IYMON4,IXMON4)
        CALL GETONE(AUX(DEN1),DD4,IYMON4,IXMON4)
        IF(STORE(KE)) CALL GETONE(KE,KE4,IYMON4,IXMON4)
        IF(STORE(EP)) CALL GETONE(EP,EP4,IYMON4,IXMON4)
        IF(STORE(AUX(VIST))) CALL GETONE(AUX(VIST),ET4,IYMON4,IXMON4)
        CALL GETONE(C1,C1C4,IYMON4,IXMON4)
```

```
       IF(STORE(C2)) CALL GETONE(C2,C2C4,IYMON4,IXMON4)
       IF(STORE(C3)) CALL GETONE(C3,C3C4,IYMON4,IXMON4)
       IF(STORE(CP)) CALL GETONE(CP,CPC4,IYMON4,IXMON4)
       IF(STORE(C11)) CALL GETONE(C11,CXC4,IYMON4,IXMON4)
       CALL GETONE(TEMP,C4C4,IYMON4,IXMON4)
       CALL GETONE(H1,H1H4,IYMON4,IXMON4)
1965 IF(IZ.NE.IZMON5) GOTO 1966
       CALL GETONE(P1,PP5,IYMON5,IXMON5)
       CALL GETONE(U1,UU5,IYMON5,IXMON5)
       CALL GETONE(V1,VV5,IYMON5,IXMON5)
       CALL GETONE(W1,WW5,IYMON5,IXMON5)
       CALL GETONE(AUX(DEN1),DD5,IYMON5,IXMON5)
       IF(STORE(KE)) CALL GETONE(KE,KE5,IYMON5,IXMON5)
       IF(STORE(EP)) CALL GETONE(EP,EP5,IYMON5,IXMON5)
       IF(STORE(AUX(VIST))) CALL GETONE(AUX(VIST),ET5,IYMON5,IXMON5)
       CALL GETONE(C1,C1C5,IYMON5,IXMON5)
       IF(STORE(C2)) CALL GETONE(C2,C2C5,IYMON5,IXMON5)
       IF(STORE(C3)) CALL GETONE(C3,C3C5,IYMON5,IXMON5)
       IF(STORE(CP)) CALL GETONE(CP,CPC5,IYMON5,IXMON5)
       IF(STORE(C11)) CALL GETONE(C11,CXC5,IYMON5,IXMON5)
       CALL GETONE(TEMP,C4C5,IYMON5,IXMON5)
       CALL GETONE(H1,H1H5,IYMON5,IXMON5)
1966 IF(IZ.NE.IZMON6) GOTO 1967
       CALL GETONE(P1,PP6,IYMON6,IXMON6)
       CALL GETONE(U1,UU6,IYMON6,IXMON6)
       CALL GETONE(V1,VV6,IYMON6,IXMON6)
       CALL GETONE(W1,WW6,IYMON6,IXMON6)
       CALL GETONE(AUX(DEN1),DD6,IYMON6,IXMON6)
       IF(STORE(KE)) CALL GETONE(KE,KE6,IYMON6,IXMON6)
       IF(STORE(EP)) CALL GETONE(EP,EP6,IYMON6,IXMON6)
       IF(STORE(AUX(VIST))) CALL GETONE(AUX(VIST),ET6,IYMON6,IXMON6)
       CALL GETONE(C1,C1C6,IYMON6,IXMON6)
       IF(STORE(C2)) CALL GETONE(C2,C2C6,IYMON6,IXMON6)
       IF(STORE(C3)) CALL GETONE(C3,C3C6,IYMON6,IXMON6)
       IF(STORE(CP)) CALL GETONE(CP,CPC6,IYMON6,IXMON6)
       IF(STORE(C11)) CALL GETONE(C11,CXC6,IYMON6,IXMON6)
       CALL GETONE(TEMP,C4C6,IYMON6,IXMON6)
       CALL GETONE(H1,H1H6,IYMON6,IXMON6)
1967 IF(IZ.NE.IZMON7) GOTO 1968
       CALL GETONE(P1,PP7,IYMON7,IXMON7)
       CALL GETONE(U1,UU7,IYMON7,IXMON7)
       CALL GETONE(V1,VV7,IYMON7,IXMON7)
       CALL GETONE(W1,WW7,IYMON7,IXMON7)
       CALL GETONE(AUX(DEN1),DD7,IYMON7,IXMON7)
       IF(STORE(KE)) CALL GETONE(KE,KE7,IYMON7,IXMON7)
       IF(STORE(EP)) CALL GETONE(EP,EP7,IYMON7,IXMON7)
       IF(STORE(AUX(VIST))) CALL GETONE(AUX(VIST),ET7,IYMON7,IXMON7)
       CALL GETONE(C1,C1C7,IYMON7,IXMON7)
       IF(STORE(C2)) CALL GETONE(C2,C2C7,IYMON7,IXMON7)
       IF(STORE(C3)) CALL GETONE(C3,C3C7,IYMON7,IXMON7)
       IF(STORE(CP)) CALL GETONE(CP,CPC7,IYMON7,IXMON7)
       IF(STORE(C11)) CALL GETONE(C11,CXC7,IYMON7,IXMON7)
       CALL GETONE(TEMP,C4C7,IYMON7,IXMON7)
       CALL GETONE(H1,H1H7,IYMON7,IXMON7)
1968 IF(IZ.NE.IZMON8) GOTO 1969
       CALL GETONE(P1,PP8,IYMON8,IXMON8)
       CALL GETONE(U1,UU8,IYMON8,IXMON8)
       CALL GETONE(V1,VV8,IYMON8,IXMON8)
       CALL GETONE(W1,WW8,IYMON8,IXMON8)
       CALL GETONE(AUX(DEN1),DD8,IYMON8,IXMON8)
```

```
        IF(STORE(KE)) CALL GETONE(KE,KE8,IYMON8,IXMON8)
        IF(STORE(EP)) CALL GETONE(EP,EP8,IYMON8,IXMON8)
        IF(STORE(AUX(VIST))) CALL GETONE(AUX(VIST),ET8,IYMON8,IXMON8)
        CALL GETONE(C1,C1C8,IYMON8,IXMON8)
        IF(STORE(C2)) CALL GETONE(C2,C2C8,IYMON8,IXMON8)
        IF(STORE(C3)) CALL GETONE(C3,C3C8,IYMON8,IXMON8)
        IF(STORE(CP)) CALL GETONE(CP,CPC8,IYMON8,IXMON8)
        IF(STORE(C11)) CALL GETONE(C11,CXC8,IYMON8,IXMON8)
        CALL GETONE(TEMP,C4C8,IYMON8,IXMON8)
        CALL GETONE(H1,H1H8,IYMON8,IXMON8)
 1969 IF(IZ.NE.IZMON9) GOTO 19691
        CALL GETONE(P1,PP9,IYMON9,IXMON9)
        CALL GETONE(U1,UU9,IYMON9,IXMON9)
        CALL GETONE(V1,VV9,IYMON9,IXMON9)
        CALL GETONE(W1,WW9,IYMON9,IXMON9)
        CALL GETONE(AUX(DEN1),DD9,IYMON9,IXMON9)
        IF(STORE(KE)) CALL GETONE(KE,KE9,IYMON9,IXMON9)
        IF(STORE(EP)) CALL GETONE(EP,EP9,IYMON9,IXMON9)
        IF(STORE(AUX(VIST))) CALL GETONE(AUX(VIST),ET9,IYMO91,IXMON9)
        CALL GETONE(C1,C1C9,IYMON9,IXMON9)
        IF(STORE(C2)) CALL GETONE(C2,C2C9,IYMON9,IXMON9)
        IF(STORE(C3)) CALL GETONE(C3,C3C9,IYMON9,IXMON9)
        IF(STORE(CP)) CALL GETONE(CP,CPC9,IYMON9,IXMON9)
        IF(STORE(C11)) CALL GETONE(C11,CXC9,IYMON9,IXMON9)
        CALL GETONE(TEMP,C4C9,IYMON9,IXMON9)
        CALL GETONE(H1,H1H9,IYMON9,IXMON9)
.9691 IF(IZ.NE.IZMON10) GOTO 19692
        CALL GETONE(P1,PP10,IYMON10,IXMON10)
        CALL GETONE(U1,UU10,IYMON10,IXMON10)
        CALL GETONE(V1,VV10,IYMON10,IXMON10)
        CALL GETONE(W1,WW10,IYMON10,IXMON10)
        CALL GETONE(AUX(DEN1),DD10,IYMON10,IXMON10)
        IF(STORE(KE)) CALL GETONE(KE,KE10,IYMON10,IXMON10)
        IF(STORE(EP)) CALL GETONE(EP,EP10,IYMON10,IXMON10)
        IF(STORE(AUX(VIST))) CALL GETONE(AUX(VIST),ET10,IYMON10,IXMON10)
        CALL GETONE(C1,C1C10,IYMON10,IXMON10)
        IF(STORE(C2)) CALL GETONE(C2,C2C10,IYMON10,IXMON10)
        IF(STORE(C3)) CALL GETONE(C3,C3C10,IYMON10,IXMON10)
        IF(STORE(CP)) CALL GETONE(CP,CPC10,IYMON10,IXMON10)
        IF(STORE(C11)) CALL GETONE(C11,CXC10,IYMON10,IXMON10)
        CALL GETONE(TEMP,C4C10,IYMON10,IXMON10)
        CALL GETONE(H1,H1H10,IYMON10,IXMON10)
.9692 CONTINUE
C
      RETURN
  197 CONTINUE
C   * ------------------- SECTION 7 ---- FINISH OF SWEEP.
C
C-pd---Printout of monitoring locations------------------------------
C
      IF(MOD(ISWEEP,TSTSWP).EQ.0.AND.IG(38).EQ.1) WRITE(6,1977)
     &   IXMON1,IYMON1,IZMON1,PP1,UU1,VV1,WW1,DD1,
     &   IXMON2,IYMON2,IZMON2,PP2,UU2,VV2,WW2,DD2,
     &   IXMON3,IYMON3,IZMON3,PP3,UU3,VV3,WW3,DD3,
     &   IXMON4,IYMON4,IZMON4,PP4,UU4,VV4,WW4,DD4,
     &   IXMON5,IYMON5,IZMON5,PP5,UU5,VV5,WW5,DD5,
     &   IXMON6,IYMON6,IZMON6,PP6,UU6,VV6,WW6,DD6,
     &   IXMON7,IYMON7,IZMON7,PP7,UU7,VV7,WW7,DD7,
     &   IXMON8,IYMON8,IZMON8,PP8,UU8,VV8,WW8,DD8,
     &   IXMON9,IYMON9,IZMON9,PP9,UU9,VV9,WW9,DD9,
     &   IXMON10,IYMON10,IZMON10,PP10,UU10,VV10,WW10,DD10
 1977 FORMAT(1X,'MONITORING VALUES :'2X,'P1',11X,'U1',11X,'V1',11X,
     &   'W1',10X,'RHO1'/,10(1X,'AT(',I2,',',I2,',',I2,'):'1P,5E13.5:,/))
      IF(MOD(ISWEEP,TSTSWP).EQ.0.AND.IG(39).EQ.1) WRITE(6,1978)
     &   IXMON1,IYMON1,IZMON1,KE1,EP1,C1C1,ET1,C4C1,
     &   IXMON2,IYMON2,IZMON2,KE2,EP2,C1C2,ET2,C4C2,
     &   IXMON3,IYMON3,IZMON3,KE3,EP3,C1C3,ET3,C4C3,
     &   IXMON4,IYMON4,IZMON4,KE4,EP4,C1C4,ET4,C4C4,
     &   IXMON5,IYMON5,IZMON5,KE5,EP5,C1C5,ET5,C4C5,
     &   IXMON6,IYMON6,IZMON6,KE6,EP6,C1C6,ET6,C4C6,
     &   IXMON7,IYMON7,IZMON7,KE7,EP7,C1C7,ET7,C4C7,
     &   IXMON8,IYMON8,IZMON8,KE8,EP8,C1C8,ET8,C4C8,
     &   IXMON9,IYMON9,IZMON9,KE9,EP9,C1C9,ET9,C4C9,
     &   IXMON10,IYMON10,IZMON10,KE10,EP10,C1C10,ET10,C4C10
 1978 FORMAT(1X,'MONITORING VALUES :'2X,'KE',11X,'EP',11X,'C1',10X,
     &   'ENUT',9X,'TEMP'/,10(1X,'LO(',I2,',',I2,',',I2,'):'1P,5E13.5:,/))
      IF(MOD(ISWEEP,TSTSWP).EQ.0.AND.IG(40).EQ.1) WRITE(6,1979)
     &   IXMON1,IYMON1,IZMON1,H1H1,C2C1,C3C1,CPC1,CXC1,
     &   IXMON2,IYMON2,IZMON2,H1H2,C2C2,C3C2,CPC2,CXC2,
     &   IXMON3,IYMON3,IZMON3,H1H3,C2C3,C3C3,CPC3,CXC3,
     &   IXMON4,IYMON4,IZMON4,H1H4,C2C4,C3C4,CPC4,CXC4,
     &   IXMON5,IYMON5,IZMON5,H1H5,C2C5,C3C5,CPC5,CXC5,
     &   IXMON6,IYMON6,IZMON6,H1H6,C2C6,C3C6,CPC6,CXC6,
     &   IXMON7,IYMON7,IZMON7,H1H7,C2C7,C3C7,CPC7,CXC7,
     &   IXMON8,IYMON8,IZMON8,H1H8,C2C8,C3C8,CPC8,CXC8,
     &   IXMON9,IYMON9,IZMON9,H1H9,C2C9,C3C9,CPC9,CXC9,
     &   IXMON10,IYMON10,IZMON10,H1H10,C2C10,C3C10,CPC10,CXC10
 1979 FORMAT(1X,'MONITORING VALUES :'2X,'H1',11X,'C2',11X,'C3',11X,
     &   'CP',10X,'SPAR'/,10(1X,'PT(',I2,',',I2,',',I2,"):'1P,5E13.5:,/))
C
C-pd---Printout heat info-------------------------------------------
C
      IF (IG(41).EQ.1) THEN
        CALL GETSOR('HEATTR1E',H1,QDOT1)
        CALL GETSOR('HEATTR1W',H1,QDOT2)
        CALL GETSOR('HEATTR1N',H1,QDOT3)
        CALL GETSOR('HEATTR1S',H1,QDOT4)
        CALL WRITBL
        CALL WRIT4R(' Qdot 1 ',QDOT1,',Qdot 2 ',QDOT2,
     &              ',Qdot 3 ',QDOT3,',Qdot 4 ',QDOT4)
      ENDIF
C
C-pd---Printout sorc and calc pumping ratio--------------------------
C
      IF(MOD(ISWEEP,NPRMON).EQ.0.OR.MOD(ISWEEP,IG(901)).EQ.0) THEN
        CALL GETSOR('XOPEN2A',R1,XMDOT2A)
        CALL GETSOR('XOPEN2B',R1,XMDOT2B)
        CALL GETSOR('XOPEN2C',R1,XMDOT2C)
        CALL GETSOR('XOPEN2D',R1,XMDOT2D)
        CALL GETSOR('XENGOUT',R1,XMDOT3)
        CALL GETSOR('XOPEN3', R1,XMDOT4)
        CALL GETSOR('XENGIN', R1,XMDOT6)
        CALL GETSOR('XOPEN1', R1,XMDOT7)
        CALL GETSOR('XENGOUT',W1,XWVEL1)
        CALL GETSOR('ZPROP',  W1,XWVEL2)
        XMDOT2=XMDOT2A+XMDOT2B+XMDOT2C+XMDOT2D
        XPR2=(-XMDOT4-XMDOT3)/XMDOT3
        XERR1=RESD(P1)*RESREF(P1)*RG(701)*100.0/XMDOT7
C       XERR2=RESD(W1)*RESREF(W1)*100.0/(XWVEL1+XWVEL1)
        XERR2=RESD(W1)*RESREF(W1)*100.0/(XWVEL1+XWVEL2)
        XFUL=(XMDOT3+XMDOT6)/RG(35)
        XMBAL=XMDOT7+XMDOT2+XMDOT4+RG(702)
        IF(MOD(ISWEEP,NPRMON).EQ.0) THEN
          CALL WRIT4R(' Mdot 1 ',XMDOT7,',Mdot 2 ',XMDOT2,
     &               ',Mdot 3 ',XMDOT4,',Mdot 4 ',XMDOT3)
          CALL WRIT1R(' PR Eng ',XPR2)
          CALL WRIT2R(' ENG  IN',XMDOT6/RG(35),',ENG OUT',XMDOT3/RG(35))
          CALL WRIT1R(' FUEL IN',XFUL)
          CALL WRIT2R(' ErrMdot',XERR1,',ErrVel ',XERR2)
          CALL WRIT1R(' SUM MAS',XMBAL)
        ENDIF
C·
        IF(MOD(ISWEEP,IG(901)).EQ.0) THEN
          CALL GETSOR('XOPEN2', H1,XEDOT2)
          CALL GETSOR('XENGOUT',H1,XEDOT3)
          CALL GETSOR('XOPEN3', H1,XEDOT4)
          CALL GETSOR('XENGIN', H1,XEDOT6)
          CALL GETSOR('XOPEN1', H1,XEDOT7)
          XECON=9.47831E-04
          CALL WRITBL
          CALL RUSHL(XMDOT7/RG(35),XMDOT2/RG(35),XMDOT4/RG(35),
     &                XMDOT6/RG(35),XMDOT3/RG(35),XFUL,XEDOT7*XECON,
     &                XEDOT2*XECON,XEDOT4*XECON,XEDOT6*XECON,
     &                XEDOT3*XECON,XPR2,XERR1,XERR2,XMBAL/RG(35))
          CALL WRITBL
        ENDIF
      ENDIF
C
C-pd---Printout max and min---------------------------------------
C
      IF(MOD(ISWEEP,NPRMON).EQ.0) THEN
        WRITE(6,*)'  P1MAX  LOC ',XP1MAX,IXPMAX,IYPMAX,IZPMAX
        WRITE(6,*)'  P1MIN  LOC ',XP1MIN,IXPMIN,IYPMIN,IZPMIN
        WRITE(6,*)'  U1MAX  LOC ',XU1MAX,IXUMAX,IYUMAX,IZUMAX
        WRITE(6,*)'  U1MIN  LOC ',XU1MIN,IXUMIN,IYUMIN,IZUMIN
        WRITE(6,*)'  V1MAX  LOC ',XV1MAX,IXVMAX,IYVMAX,IZVMAX
        WRITE(6,*)'  V1MIN  LOC ',XV1MIN,IXVMIN,IYVMIN,IZVMIN
        WRITE(6,*)'  W1MAX  LOC ',XW1MAX,IXWMAX,IYWMAX,IZWMAX
        WRITE(6,*)'  W1MIN  LOC ',XW1MIN,IXWMIN,IYWMIN,IZWMIN
        WRITE(6,*)'  H1MAX  LOC ',XH1MAX,IXHMAX,IYHMAX,IZHMAX
        WRITE(6,*)'  H1MIN  LOC ',XH1MIN,IXHMIN,IYHMIN,IZHMIN
        WRITE(6,*)'  T1MAX  LOC ',XT1MAX,IXTMAX,IYTMAX,IZTMAX
        WRITE(6,*)'  T1MIN  LOC ',XT1MIN,IXTMIN,IYTMIN,IZTMIN
        WRITE(6,*)'  KEMAX  LOC ',XKEMAX,IXKMAX,IYKMAX,IZKMAX
C       WRITE(6,*)'  KEMIN  LOC ',XKEMIN,IXKMIN,IYKMIN,IZKMIN
        WRITE(6,*)'  EPMAX  LOC ',XEPMAX,IXEMAX,IYEMAX,IZEMAX
C       WRITE(6,*)'  EPMIN  LOC ',XEPMIN,IXEMIN,IYEMIN,IZEMIN
        WRITE(6,*)'  ETMAX  LOC ',XETMAX,IXXMAX,IYXMAX,IZXMAX
C       WRITE(6,*)'  ETMIN  LOC ',XETMIN,IXXMIN,IYXMIN,IZXMIN
      ENDIF
      IF(ISWEEP.EQ.FSWEEP+2) NPRMON=INPR
      IF(MOD(ISWEEP,TSTSWP).NE.0) WRITE(6,*)' ISWEEP = ',ISWEEP
C
C-pd---Printout heat total----------------------------------------
C
      IF(ISWEEP.EQ.LSWEEP.OR.MOD(ISWEEP,IG(901)).EQ.0) THEN
        CALL WRITBL
        CALL WRIT1R('Qdot,Tot',QDTTOT)
        CALL WRIT4R(' QTOT 1 ',QDOT01,',QTOT 2 ',QDOT02,
     &              ',QTOT 3 ',QDOT03,',QTOT 4 ',QDOT04)
```

```
        QDTTOT=0.0
        QDOT01=0.0
        QDOT02=0.0
        QDOT03=0.0
        QDOT04=0.0
        CALL WRITBL
      ENDIF
C
C-pd---Check to stop run------------------------------------------
C
      INQUIRE(FILE='ABORT',EXIST=LSG1)
      IF(LSG1) THEN
        OPEN(91,FILE='ABORT')
        CLOSE(91,STATUS='DELETE')
        LSWEEP=ISWEEP+2
        WRITE(6,*)' ==> ABORT CALLED:  STOP IN 2 SWEEPS '
        LSG1=.FALSE.
      ENDIF
C
C-pd---Modify relaxation without killing run-----------------------
C
      INQUIRE(FILE='RELAXP',EXIST=LSG2)
      IF(LSG2) THEN
        OPEN(92,FILE='RELAXP')
        WRITE(6,*)' ==> MODIFYING RELAX P1          OLD VALVE=',
     &             DTFALS(P1)
        READ(92,1971)XRELP1
        DTFALS(P1)=XRELP1
        CLOSE(92,STATUS='DELETE')
        WRITE(6,*)' ==>             ISWEEP    &     NEW VALVE=',
     &             DTFALS(P1),ISWEEP
        ITST=TSTSWP
        TSTSWP=1
        IOPEN=1
        LSG2=.FALSE.
      ENDIF
C
      INQUIRE(FILE='RELAXT',EXIST=LSG3)
      IF(LSG3) THEN
        OPEN(93,FILE='RELAXT')
        WRITE(6,*)' ==> MODIFYING RELAX KE & EP    OLD VALVES=',
     &             DTFALS(KE),DTFALS(EP)
        READ(93,1972)XRELKE,XRELEP
        DTFALS(KE)=XRELKE
        DTFALS(EP)=XRELEP
        CLOSE(93,STATUS='DELETE')
        WRITE(6,*)' ==>             ISWEEP    &    NEW VALVES=',
     &             DTFALS(KE),DTFALS(EP),ISWEEP
        IF(IOPEN.EQ.0) THEN
          ITST=TSTSWP
          TSTSWP=1
          IOPEN=1
        ENDIF
        LSG3=.FALSE.
      ENDIF
C
      INQUIRE(FILE='RELAXS',EXIST=LSG4)
      IF(LSG4) THEN
        OPEN(94,FILE='RELAXS')
        IF(SOLVE(C2)) THEN
        WRITE(6,*)' ==> MODIFYING RELAX H1 C1 & C2 OLD VALVES=',
     &             DTFALS(H1),DTFALS(C1),DTFALS(C2)
          READ(94,1973)XRELH1,XRELC1,XRELC2
          DTFALS(H1)=XRELH1
          DTFALS(C1)=XRELC1
          DTFALS(C2)=XRELC2
        WRITE(6,*)' ==>             ISWEEP    &    NEW VALVES=',
     &            DTFALS(H1),DTFALS(C1),DTFALS(C2),ISWEEP
        ELSE
        WRITE(6,*)' ==> MODIFYING RELAX H1 & C1    OLD VALVES=',
     &             DTFALS(H1),DTFALS(C1)
          READ(94,1972)XRELH1,XRELC1
          DTFALS(H1)=XRELH1
          DTFALS(C1)=XRELC1
        WRITE(6,*)' ==>             ISWEEP    &    NEW VALVES=',
     &             DTFALS(H1),DTFALS(C1),ISWEEP
        ENDIF
        CLOSE(94,STATUS='DELETE')
        IF(IOPEN.EQ.0) THEN
          ITST=TSTSWP
          TSTSWP=1
          IOPEN=1
        ENDIF
        LSG4=.FALSE.
      ENDIF
C
      INQUIRE(FILE='RELAXV',EXIST=LSG5)
      IF(LSG5) THEN
        OPEN(95,FILE='RELAXV')
        WRITE(6,*)' ==> MODIFYING RELAX U1 V1 & W1 OLD VALVES=',
     &             DTFALS(U1),DTFALS(V1),DTFALS(W1)
        READ(95,1973)XRELU1,XRELV1,XRELW1
        DTFALS(U1)=XRELU1
        DTFALS(V1)=XRELV1
        DTFALS(W1)=XRELW1
        WRITE(6,*)' ==>             ISWEEP    &    NEW VALVES=',
     &             DTFALS(U1),DTFALS(V1),DTFALS(W1),ISWEEP
        CLOSE(95,STATUS='DELETE')
        IF(IOPEN.EQ.0) THEN
          ITST=TSTSWP
          TSTSWP=1
          IOPEN=1
        ENDIF
        LSG5=.FALSE.
      ENDIF
C
      INQUIRE(FILE='DUMPIT',EXIST=LSG6)
      IF(LSG6) THEN
        OPEN(96,FILE='DUMPIT')
        CLOSE(96,STATUS='DELETE')
        CALL AUTCHA(ISWEEP)
        LSG6=.FALSE.
      ELSEIF (MOD(ISWEEP,IG(902)).EQ.0) THEN
        CALL AUTCHA(ISWEEP)
      ENDIF
C
      INQUIRE(FILE='TSTMOD',EXIST=LSG7)
      IF(LSG7) THEN
        OPEN(97,FILE='TSTMOD')
        WRITE(6,*)' ==> MODIFYING TSTSWP            OLD VALVE=',
     &             TSTSWP
        READ(97,1974)TSTSWP
        IF(IOPEN.EQ.0) THEN
          ITST=TSTSWP
          IOPEN=1
        ENDIF
        WRITE(6,*)' ==>             ISWEEP    &     NEW VALVE=',
     &             TSTSWP,ISWEEP
        CLOSE(97,STATUS='DELETE')
        LSG7=.FALSE.
      ENDIF
C
      INQUIRE(FILE='NPRMOD',EXIST=LSG8)
      IF(LSG8) THEN
        OPEN(98,FILE='NPRMOD')
        WRITE(6,*)' ==> MODIFYING NPRMON           OLD VALVE=',
     &             NPRMON
        READ(98,1974)NPRMON
        WRITE(6,*)' ==>             ISWEEP    &     NEW VALVE=',
     &             NPRMON,ISWEEP
        CLOSE(98,STATUS='DELETE')
        LSG8=.FALSE.
      ENDIF
C
      INQUIRE(FILE='IGGMOD',EXIST=LSG9)
      IF(LSG9) THEN
        OPEN(99,FILE='IGGMOD')
        WRITE(6,*)' ==> MODIFYING IG(38-41)       OLD VALVES=',
     &             IG(38),IG(39),IG(40),IG(41)
        READ(99,1975)IG(38),IG(39),IG(40),IG(41)
        WRITE(6,*)' ==>             ISWEEP    &    NEW VALVES=',
     &             IG(38),IG(39),IG(40),IG(41),ISWEEP
        CLOSE(99,STATUS='DELETE')
        LSG9=.FALSE.
      ENDIF
C
      INQUIRE(FILE='ML2MOD',EXIST=LSG9)
      IF(LSG9) THEN
        OPEN(100,FILE='ML2MOD')
        WRITE(6,*)' ==> MODIFYING IXYZMON2        OLD VALVES=',
     &             IXMON2,IYMON2,IZMON2
        READ(100,1976)IXMON2,IYMON2,IZMON2
        WRITE(6,*)' ==>             ISWEEP    &    NEW VALVES=',
     &             IXMON2,IYMON2,IZMON2,ISWEEP
        CLOSE(100,STATUS='DELETE')
        LSG9=.FALSE.
      ENDIF
C
      INQUIRE(FILE='ML3MOD',EXIST=LSG9)
      IF(LSG9) THEN
        OPEN(101,FILE='ML3MOD')
        WRITE(6,*)' ==> MODIFYING IXYZMON3        OLD VALVES=',
     &             IXMON3,IYMON3,IZMON3
        READ(101,1976)IXMON3,IYMON3,IZMON3
        WRITE(6,*)' ==>             ISWEEP    &    NEW VALVES=',
     &             IXMON3,IYMON3,IZMON3,ISWEEP
        CLOSE(101,STATUS='DELETE')
        LSG9=.FALSE.
      ENDIF
C
```

```
      INQUIRE(FILE='RAXVMD',EXIST=LSG9)
      IF(LSG9) THEN
        OPEN(102,FILE='RAXVMD')
        WRITE(6,*)' ==> READING MODIFICATION FOR RAX VEL '
        READ(102,1971)RAXFTV
        WRITE(6,*)' ==>                ISWEEP    &          FACTOR=',
     &              RAXFTV,ISWEEP
        IRAXV=1
        CLOSE(102,STATUS='DELETE')
        LSG9=.FALSE.
      ENDIF
C
      INQUIRE(FILE='RAXTMD',EXIST=LSG9)
      IF(LSG9) THEN
        OPEN(102,FILE='RAXTMD')
        WRITE(6,*)' ==> READING MODIFICATION FOR RAX TURB '
        READ(102,1971)RAXFTT
        WRITE(6,*)' ==>                ISWEEP    &          FACTOR=',
     &              RAXFTT,ISWEEP
        IRAXT=1
        CLOSE(102,STATUS='DELETE')
        LSG9=.FALSE.
      ENDIF
C
      INQUIRE(FILE='RAXSMD',EXIST=LSG9)
      IF(LSG9) THEN
        OPEN(102,FILE='RAXSMD')
        WRITE(6,*)' ==> READING MODIFICATION FOR RAX SCAL '
        READ(102,1971)RAXFTS
        WRITE(6,*)' ==>                ISWEEP    &          FACTOR=',
     &              RAXFTS,ISWEEP
        IRAXS=1
        CLOSE(102,STATUS='DELETE')
        LSG9=.FALSE.
      ENDIF
C
 1971 FORMAT(F12.8)
 1972 FORMAT(2F12.8)
 1973 FORMAT(3F12.8)
 1974 FORMAT(I5)
 1975 FORMAT(4I2)
 1976 FORMAT(3I3)
C
      RETURN
  198 CONTINUE
C   * ------------------- SECTION 8 ---- FINISH OF TIME STEP.
      RETURN
C***********************************************************************
C
C--- GROUP 20. Preliminary print-out
C
   20 CONTINUE
      RETURN
C***********************************************************************
C   * Make changes for this group only in group 19.
C--- GROUP 21. Print-out of variables
C--- GROUP 22. Spot-value print-out
C***********************************************************************
C
C--- GROUP 23. Field print-out and plot control
```

```
   23 CONTINUE
      RETURN
C***************************************************************
C
C--- GROUP 24. Dumps for restarts
C
   24 CONTINUE
      RETURN
      END
C***********************************************************************
      SUBROUTINE TEMPER(HSTAT,T0,T,CPDR,RGAS,SC,NSC,NFO)
C***********************************************************************
C  TEMPER uses an iterative procedure to calculate temperature
C  given H1 and a guess for temperature
C---------------------------------------------------------------------
C
      DIMENSION SC(NSC)
      DATA NITER,DT0,TMIN/12,50.,12.345/
C
      CALL ENTHAL(T0,HHH,CPDR,SC,NSC,NFO)
C
      CP=CPDR*RGAS
      ENTH=CP*T0
      DT=(HSTAT-ENTH)/(CP+1.E-15)
      TEMPL=T0
      IF(NFO.GE.4) WRITE(6,900) T0,ENTH,HSTAT,RGAS,SC(1),SC(2),SC(3)
      TEMP =T0+DT
      ITER=0
  100 ENTHL=ENTH
      ITER=ITER+1
      CALL ENTHAL(TEMP,HHH,CPDR,SC,NSC,NFO)
      ENTH=CPDR*RGAS*TEMP
      RENTH=(HSTAT-ENTHL)/((ENTH-ENTHL) +1.E-9)
      IF(NFO.GE.4) WRITE(6,910) ITER,TEMP,ENTH,ENTHL,HSTAT,RENTH
      IF(ABS(ENTH-ENTHL).LT..001*ABS(ENTH)) RENTH=1.
      TEMP1=TEMPL+(TEMP-TEMPL)*RENTH
      TEMP1=AMAX1(TEMP1,.5*TEMP,TMIN)
      TEMP1=AMIN1(TEMP1,1.5*TEMP,5000.)
      TEMPL=TEMP
      TEMP=TEMP1
      AR=ABS(RENTH)
      IF( (AR.GT.1.005 .OR. AR.LT..995) .AND. ITER.LT.NITER) GO TO 100
      T=TEMP
      RETURN
C
  900 FORMAT(' T0 E HS RG SC',1P,7E12.4)
  910 FORMAT(' IT T E EL HS RE',I3,1P,5E12.4)
C
      END
C
C***********************************************************************
      SUBROUTINE ENTHAL(TEMP,HSUM,CPSUM,SC,NS,NFO)
C***********************************************************************
C  ENTHAL calculates H/RT from JANNAF data.  The order of
C  species is N O C H.
C---------------------------------------------------------------------
C
      DIMENSION SC(*),ZS(7,2,4)
      DATA ZS/ 0.28532899E+01,  0.16022128E-02, -0.62936893E-06,
     &         0.11441022E-09, -0.78057465E-14, -0.89008093E+03,
     &          0.63964897E+01,  0.37044177E+01, -0.14218753E-02,
     &          0.28670392E-05, -0.12028885E-08, -0.13954677E-13,
     &         -0.10640795E+04,  0.22336285E+01,
     &          0.36122139E+01,  0.74853166E-03, -0.19820647E-06,
     &          0.33749008E-10, -0.23907374E-14, -0.11978151E+04,
     &          0.36703307E+01,  0.37837135E+01, -0.30233634E-02,
     &          0.99492751E-05, -0.98189101E-08,  0.33031825E-11,
     &         -0.10638107E+04,  0.36416345E+01,
     &          0.44608041E+01,  0.30981719E-02, -0.12392571E-05,
     &          0.22741325E-09, -0.15525954E-13, -0.48961442E+05,
     &         -0.98635982E+00,  0.24007797E+01,  0.87350957E-02,
     &         -0.66070878E-05,  0.20021861E-08,  0.63274039E-15,
     &         -0.48377527E+05,  0.96951457E+01,
     &          0.27167633E+01,  0.29451374E-02, -0.80224374E-06,
     &          0.10226682E-09, -0.48472145E-14, -0.29905826E 05,
     &          0.66305671E+01,  0.40701275E+01, -0.11084499E-02,
     &          0.41521180E-05, -0.29637404E-08,  0.80702103E-12,
     &         -0.30279722E+05, -0.32270046E+00 /
C
      K=1
      IF(TEMP.LT.1000.) K=2
      TEMP2=TEMP*TEMP
      HSUM=0.
      CPSUM=0.
      DO 100 IS=1,NS
      CP1=ZS(1,K,IS)
      CP2=ZS(2,K,IS)*TEMP
      CP3=ZS(3,K,IS)*TEMP2
      CP4=ZS(4,K,IS)*TEMP2*TEMP
      CP5=ZS(5,K,IS)*TEMP2*TEMP2
      CPSUM=CPSUM+SC(IS)*(CP1+CP2+CP3+CP4+CP5)
  100 HSUM =HSUM+
     1  SC(IS)*(CP1+.5*CP2+.33333*CP3+.25*CP4+.2*CP5+ZS(6,K,IS)/TEMP)
C
      RETURN
      END
C
C***********************************************************************
C     SUBROUTINE XGETCV(N,M,C,V)
C***********************************************************************
C  XGETCV used to set up procedure to get a patch co and val.
C----------------------------------------------------------------------
C
C     COMMON/IDATA/IDFIL1(70),NUMREG,IDFIL2(49)
C     COMMON/NPAT/NAMPAT(100)
C     CHARACTER N*(*),NAMPAT*8
C
C     IR=IRPAT(N)
C     CALL XCV(IR,M,C,V)

      RETURN
C     END
C
 ***********************************************************************
C     SUBROUTINE XCV(IR,MPHID,C,V)
C***********************************************************************
   XCV used to get a patch co and val.
C----------------------------------------------------------------------
C
      COMMON F(1)
```

```
C      COMMON/ICOVL/M04,IOPHI
C      LOGICAL QLT
C      INCLUDE 'SATEAR'
C
C      MPHI=MPHID
C      IO=0
C          IF(EARTH) IO=IORTCV
C          IF(QLT(F(IO+10*IR-8),23.0).AND.MPHI.LE.2) MPHI=MPHI+8
C      IOPHI=IORCV(MPHI)
C          IF(IOPHI.EQ.IO+NRTCV) GO TO 5
C      IOPHI=IOPHI-4
C        DO 2 I=1,NUMREG
C      IOPHI=IOPHI+4
C      IOL=IORCVL(MPHI)
C          IF(EARTH) IOL=IOL+IORCVF(MPHI)-4
C          IF(IOPHI.EQ.IOL+4) GO TO 5
C          IF(IABS(IFIX(F(IOPHI+1))).NE.IR) GO TO 2
C      C=F(IOPHI+2)
C      V=F(IOPHI+3)
C      GO TO 7
C    2   CONTINUE
C    5 C=-999.0
C      V=0.0
C    7 CONTINUE
C
C      RETURN
C      END
C
C*************************************************************************
C      SUBROUTINE XSETCV(N,M,C,V,CF,VF)
C*************************************************************************
C  XGETCV used to set up procedure to modify a patch co and val.
C-----------------------------------------------------------------------
C
C      COMMON/IDATA/IDFIL1(70),NUMREG,IDFIL2(49)
C      COMMON/NPAT/NAMPAT(100)
C      CHARACTER N*(*),NAMPAT*8
C
C      IR=IRPAT(N)
C      CALL XSCV(IR,M,C,V,CF,VF)
C
C
C      RETURN
C      END
C
C*************************************************************************
C      SUBROUTINE XSCV(IR,MPHID,C,V,CF,VF)
C*************************************************************************
C  XCV used to get a patch co and val.
C-----------------------------------------------------------------------
C
C      COMMON F(1)
C      COMMON/ICOVL/M04,IOPHI
C      LOGICAL QLT
C      INCLUDE 'SATEAR'
C
C      MPHI=MPHID
C      IO=0
C          IF(EARTH) IO=IORTCV
C          IF(QLT(F(IO+10*IR-8),23.0).AND.MPHI.LE.2) MPHI=MPHI+8
```

```
C     IOPHI=IORCV(MPHI)
C         IF(IOPHI.EQ.IO+NRTCV) GO TO 5
C     IOPHI=IOPHI-4
C       DO 2 I=1,NUMREG
C     IOPHI=IOPHI+4
C     IOL=IORCVL(MPHI)
C         IF(EARTH) IOL=IOL+IORCVF(MPHI)-4
C         IF(IOPHI.EQ.IOL+4) GO TO 5
C         IF(IABS(IFIX(F(IOPHI+1))).NE.IR) GO TO 2
C     C=F(IOPHI+2)
C     V=F(IOPHI+3)
C     WRITE(6,*)' IN SETCV   VAR & OLD VALUES= ',MPHI,C,V
C     F(IOPHI+2)=F(IOPHI+2)*CF
C     F(IOPHI+3)=F(IOPHI+3)*VF
C     C=F(IOPHI+2)
C     V=F(IOPHI+3)
C     WRITE(6,*)' IN SETCV   VAR & NEW VALUES= ',MPHI,C,V
C     GO TO 7
C   2   CONTINUE
C   5 C=-999.0
C     V=0.0
C   7 CONTINUE
C
C     RETURN
C     END
C
C***********************************************************************
      SUBROUTINE RUSHL(XMD1,XMD2,XMD3,XMD4,XMD5,XMD8,XEG1,XEG2,XEG3,
     &                 XEG4,XEG5,XPR1,XEM,XEV,XSM)
C***********************************************************************
C  RUSHL prints flow rate and convergence info
C-----------------------------------------------------------------------
C
      WRITE(6,*)' ***************************************************'
      WRITE(6,*)' *******   FLOW & CONVERGENCE DATA             *******'
      WRITE(6,*)' ***************************************************'
      WRITE(6,101)XMD1
      WRITE(6,102)XMD2
      WRITE(6,103)XMD3
      WRITE(6,104)XMD4
      WRITE(6,105)XMD5
      WRITE(6,108)XMD8
      WRITE(6,109)XEG1
      WRITE(6,110)XEG2
      WRITE(6,111)XEG3
      WRITE(6,112)XEG4
      WRITE(6,113)XEG5
      WRITE(6,116)XPR1
      WRITE(6,118)XEM
      WRITE(6,119)XEV
      WRITE(6,120)XSM
      WRITE(6,*)' ***************************************************'
  101 FORMAT('   MASS FLOW FRONT BAFFLES     ',F12.4,' LB/S ')
  102 FORMAT('   MASS FLOW BACK BAFFLES      ',F12.4,' LB/S ')
  103 FORMAT('   MASS FLOW CHIMNEY BAFFLES   ',F12.4,' LB/S ')
  104 FORMAT('   MASS FLOW INTO ENGINE       ',F12.4,' LB/S ')
  105 FORMAT('   MASS FLOW OUT OF ENGINE     ',F12.4,' LB/S ')
  108 FORMAT('   MASS FLOW OF FUEL           ',F12.4,' LB/S ')
  109 FORMAT('   ENERGY FLOW FRONT BAFFLES   ',F12.4,' BTU/S ')
  110 FORMAT('   ENERGY FLOW BACK BAFFLES    ',F12.4,' BTU/S ')
  111 FORMAT('    ENERGY FLOW CHIMNEY BAFFLES ',F12.4,' BTU/S ')
  112 FORMAT('    ENERGY FLOW INTO ENGINE     ',F12.4,' BTU/S ')
  113 FORMAT('    ENERGY FLOW OUT OF ENGINE   ',F12.4,' BTU/S ')
  116 FORMAT('    ENGINE PUMPING RATIO        ',F12.4,)
  118 FORMAT('    NORMALIZED MASS ERROR       ',F12.4,' %     ')
  119 FORMAT('    NORMALIZED MOMENTUM ERROR   ',F12.4,' %     ')
  120 FORMAT('    SUM OF ALL MASS             ',F12.4,' LB/S ')
      RETURN
      END
C
C*********************************************************************
      SUBROUTINE AUTCHA(ISW)
C*********************************************************************
C  AUTUCH writes phida file.
C-----------------------------------------------------------------
      DIMENSION JDATE(6)
C
      CALL DUMP
C-pd---WARNING:  The following two calls may be machine dependent-----
      CALL IDATE(JDATE(1))
      CALL ITIME(JDATE(4))
      WRITE(6,*)' **** DUMP CALLED ****    ISWEEP=',ISW
      WRITE(6,*)'  DAY  MONTH  YEAR    +++    HOUR  MINUTE  SECOND'
      WRITE(6,1974)JDATE
 1974 FORMAT(I4,I6,I8,8X,I6,I7,I8)
C-----------------------------------------------------------------
C
      RETURN
      END
```

DISTRIBUTION QUESTIONNAIRE

The Naval Civil Engineering Laboratory is revising its primary distribution lists.

SUBJECT CATEGORIES

1 **SHORE FACILITIES**
1A Construction methods and materials (including corrosion control, coatings)
1B Waterfront structures (maintenance/deterioration control)
1C Utilities (including power conditioning)
1D Explosives safety
1E Aviation Engineering Test Facilities
1F Fire prevention and control
1G Antenna technology
1H Structural analysis and design (including numerical and computer techniques)
1J Protective construction (including hardened shelters, shock and vibration studies)
1K Soil/rock mechanics
1L Airfields and pavements
1M Physical security
2 **ADVANCED BASE AND AMPHIBIOUS FACILITIES**
2A Base facilities (including shelters, power generation, water supplies)
2B Expedient roads/airfields/bridges
2C Over-the-beach operations (including breakwaters, wave forces)
2D POL storage, transfer, and distribution
2E Polar engineering
3 **ENERGY/POWER GENERATION**
3A Thermal conservation (thermal engineering of buildings, HVAC systems, energy loss measurement, power generation)
3B Controls and electrical conservation (electrical systems, energy monitoring and control systems)
3C Fuel flexibility (liquid fuels, coal utilization, energy from solid waste)
3D Alternate energy source (geothermal power, photovoltaic power systems, solar systems, wind systems, energy storage systems)
3E Site data and systems integration (energy resource data, integrating energy systems)
3F EMCS design
4 **ENVIRONMENTAL PROTECTION**
4A Solid waste management
4B Hazardous/toxic materials management
4C Waterwaste management and sanitary engineering
4D Oil pollution removal and recovery
4E Air pollution
4F Noise abatement
5 **OCEAN ENGINEERING**
5A Seafloor soils and foundations
5B Seafloor construction systems and operations (including diver and manipulator tools)
5C Undersea structures and materials
5D Anchors and moorings
5E Undersea power systems, electromechanical cables, and connectors
5F Pressure vessel facilities
5G Physical environment (including site surveying)
5H Ocean-based concrete structures
5J Hyperbaric chambers
5K Undersea cable dynamics
ARMY FEAP
BDG Shore Facilities
NRG Energy
ENV Environmental/Natural Responses
MGT Management
PRR Pavements/Railroads

TYPES OF DOCUMENTS

D – Techdata Sheets; **R** – Technical Reports and Technical Notes; **G** – NCEL Guides and Abstracts; **I** – Index to TDS; **U** – User Guides; □ *None - remove my name*

Old Address:

____

____

____

____

Telephone No.: ________________

New Address:

____

____

____

____

Telephone No.: ________________

INSTRUCTIONS

The Naval Civil Engineering Laboratory has revised its primary distribution lists. To help us verify our records and update our data base, please do the following:

- Add - circle number on list
- Remove my name from all your lists - check box on list.
- Change my address - line out incorrect line and write in correction **(DO NOT REMOVE LABEL).**
- Number of copies should be entered after the title of the subject categories you select.
- Are we sending you the correct type of document? If not, circle the type(s) of document(s) you want to receive listed on the back of this card.

Fold on line, staple, and drop in mail.

DEPARTMENT OF THE NAVY
Naval Civil Engineering Laboratory
560 Laboratory Drive
Port Hueneme CA 93043-4328

Official Business
Penalty for Private Use, $300

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD
FIRST CLASS PERMIT NO. 12503 WASH D.C.

POSTAGE WILL BE PAID BY ADDRESSEE

COMMANDING OFFICER
CODE L34
560 LABORATORY DRIVE
NAVAL CIVIL ENGINEERING LABORATORY
PORT HUENEME CA 93043-4328

NCEL DOCUMENT EVALUATION

You are number one with us; how do we rate with you?

We at NCEL want to provide you our customer the best possible reports but we need your help. Therefore, I ask you to please take the time from your busy schedule to fill out this questionnaire. Your response will assist us in providing the best reports possible for our users. I wish to thank you in advance for your assistance. I assure you that the information you provide will help us to be more responsive to your future needs.

R. N. STORER, Ph.D, P.E.
Technical Director

DOCUMENT NO. ________ TITLE OF DOCUMENT: ______________________________

____

Date: __________ Respondent Organization : ______________________________

Name: ______________________________ Activity Code: ____________________

Phone: ______________________________ Grade/Rank: ____________________

Category (*please check*):

Sponsor _____ User _____ Proponent _____ Other (*Specify*) ____________________

Please answer on your behalf only; not on your organization's. Please check (*use an X*) only the block that most closely describes your attitude or feeling toward that statement:

SA Strongly Agree A Agree O Neutral D Disagree SD Strongly Disagree

		SA	A	N	D	SD
1.	The technical quality of the report is comparable to most of my other sources of technical information.	( )	( )	( )	( )	( )
2.	The report will make significant improvements in the cost and or performance of my operation.	( )	( )	( )	( )	( )
3.	The report acknowledges related work accomplished by others.	( )	( )	( )	( )	( )
4.	The report is well formatted.	( )	( )	( )	( )	( )
5.	The report is clearly written.	( )	( )	( )	( )	( )
6.	The conclusions and recommendations are clear and directly supported by the contents of the report.	( )	( )	( )	( )	( )
7.	The graphics, tables, and photographs are well done.	( )	( )	( )	( )	( )

Do you wish to continue getting NCEL reports? □ YES □ NO

Please add any comments (e.g., in what ways can we improve the quality of our reports?) on the back of this form.

Comments:

Fold on line, staple, and drop in mail.

DEPARTMENT OF THE NAVY
Naval Civil Engineering Laboratory
560 Laboratory Drive
Port Hueneme CA 93043-4328

Official Business
Penalty for Private Use, $300

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD
FIRST CLASS PERMIT NO. 12503 WASH D.C.

POSTAGE WILL BE PAID BY ADDRESSEE

COMMANDING OFFICER
CODE L03
560 LABORATORY DRIVE
NAVAL CIVIL ENGINEERING LABORATORY
PORT HUENEME CA 93043-4328